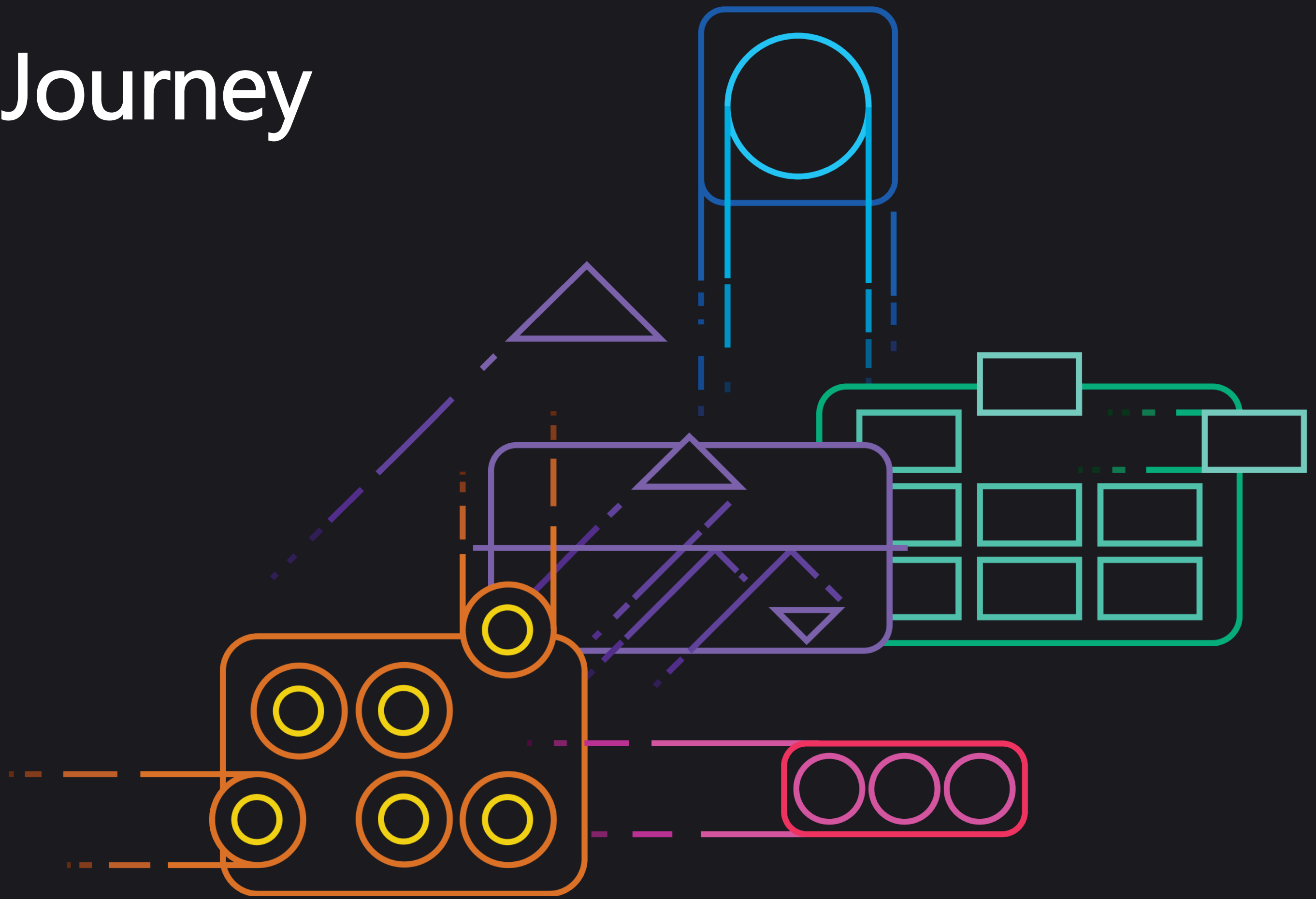
The Microsoft DevOps Journey so far...

Sasha Rosenbaum
@DivineOps



Sasha Rosenbaum

Sr. Program Manager

@GitHub

@DivineOps

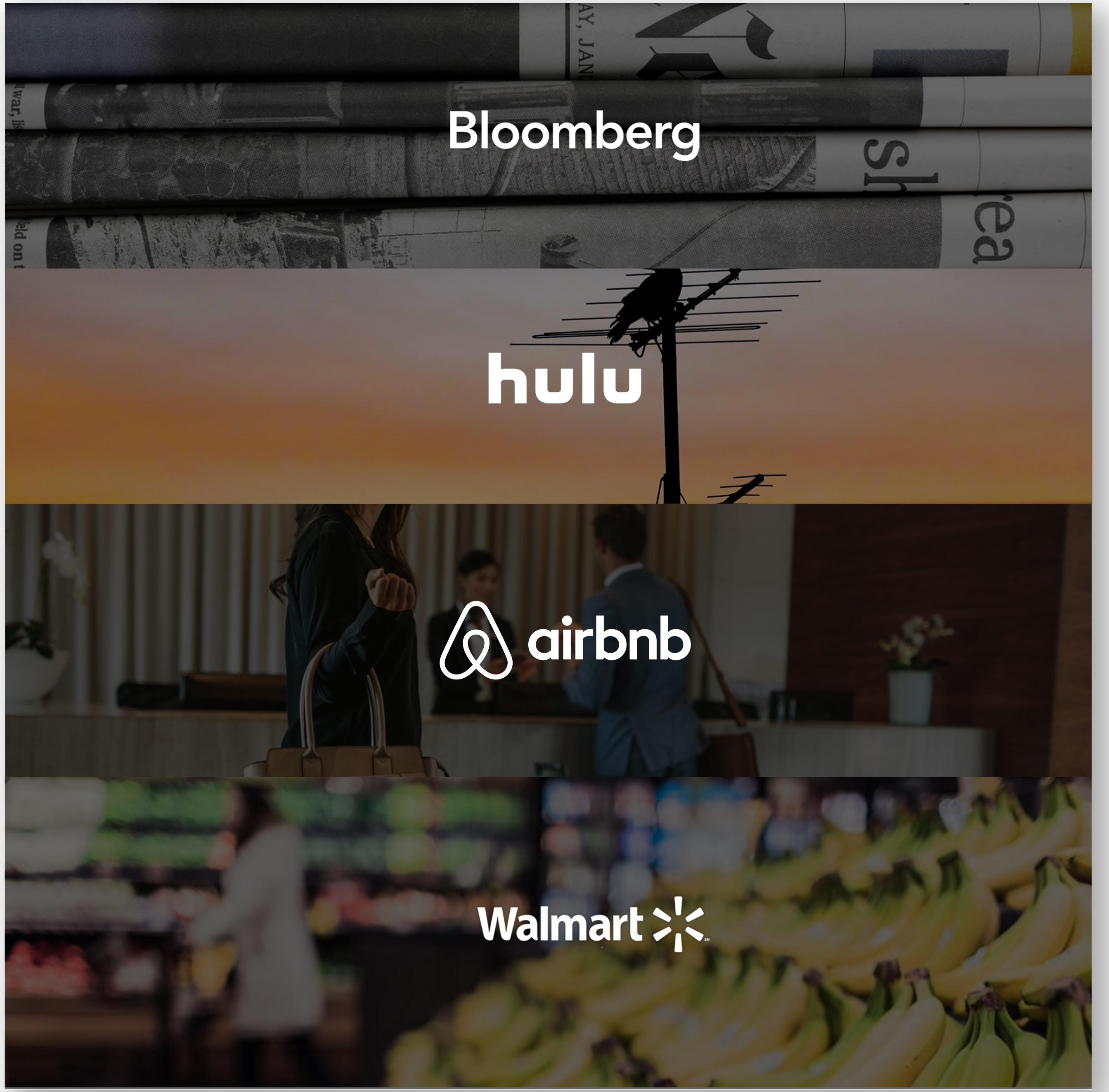
@DivineOps

And you?

Why DevOps?

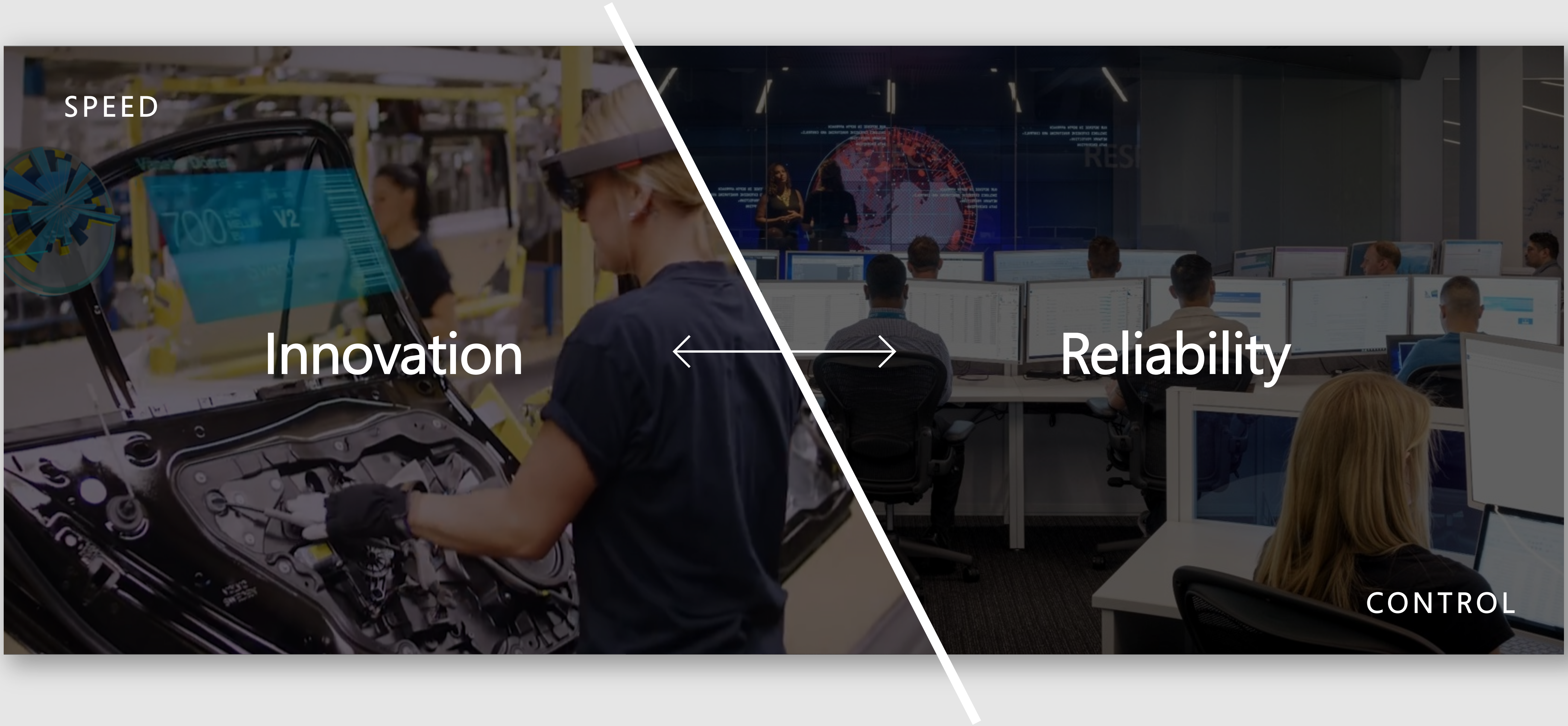
Every company is becoming a software company

62% of CEOs have an initiative to make their businesses more digital



Software delivery paradox

Speed vs. control



Effective DevOps

Key practices

Culture

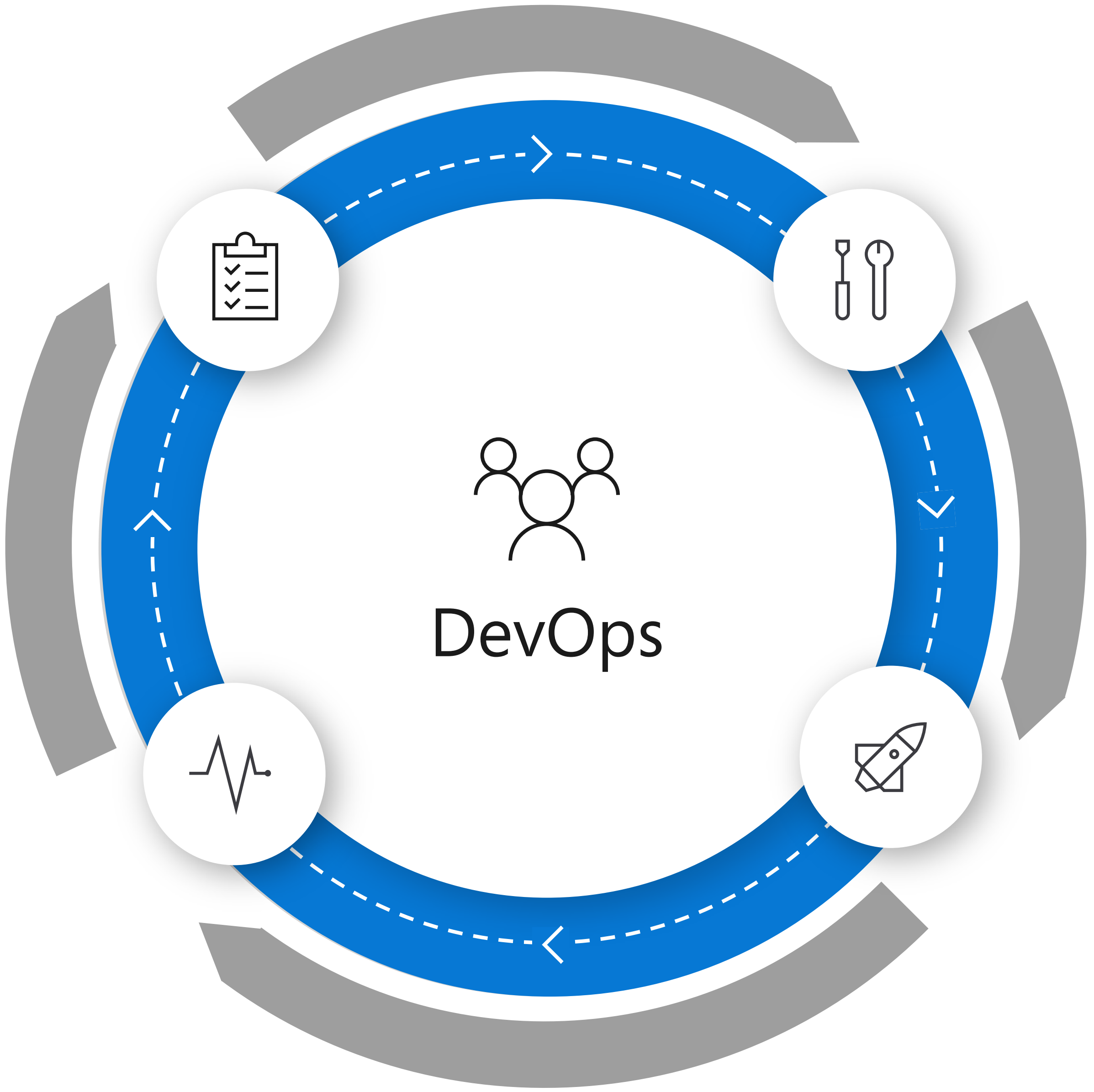
Automation

Lean

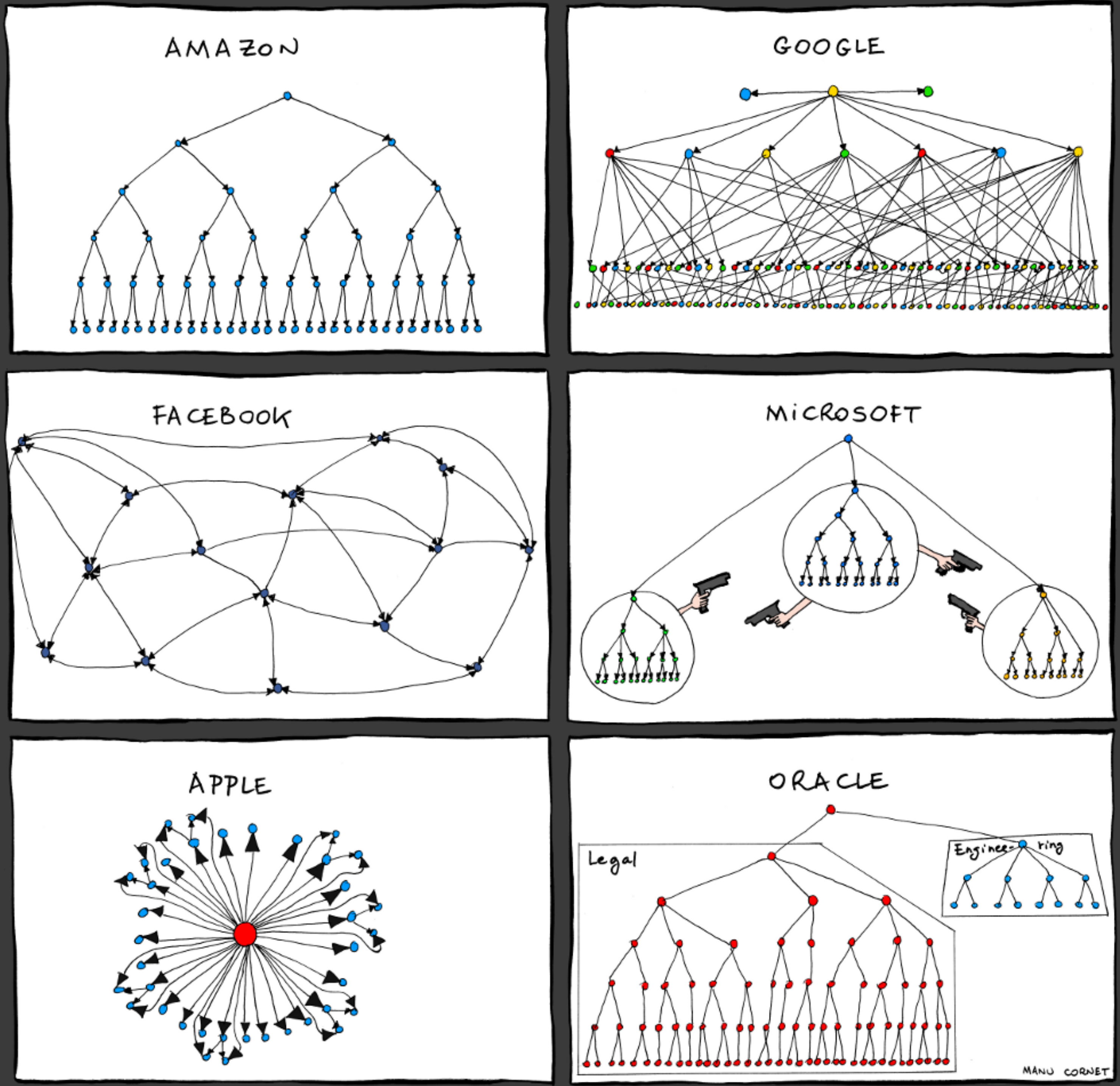
Measurement

Sharing

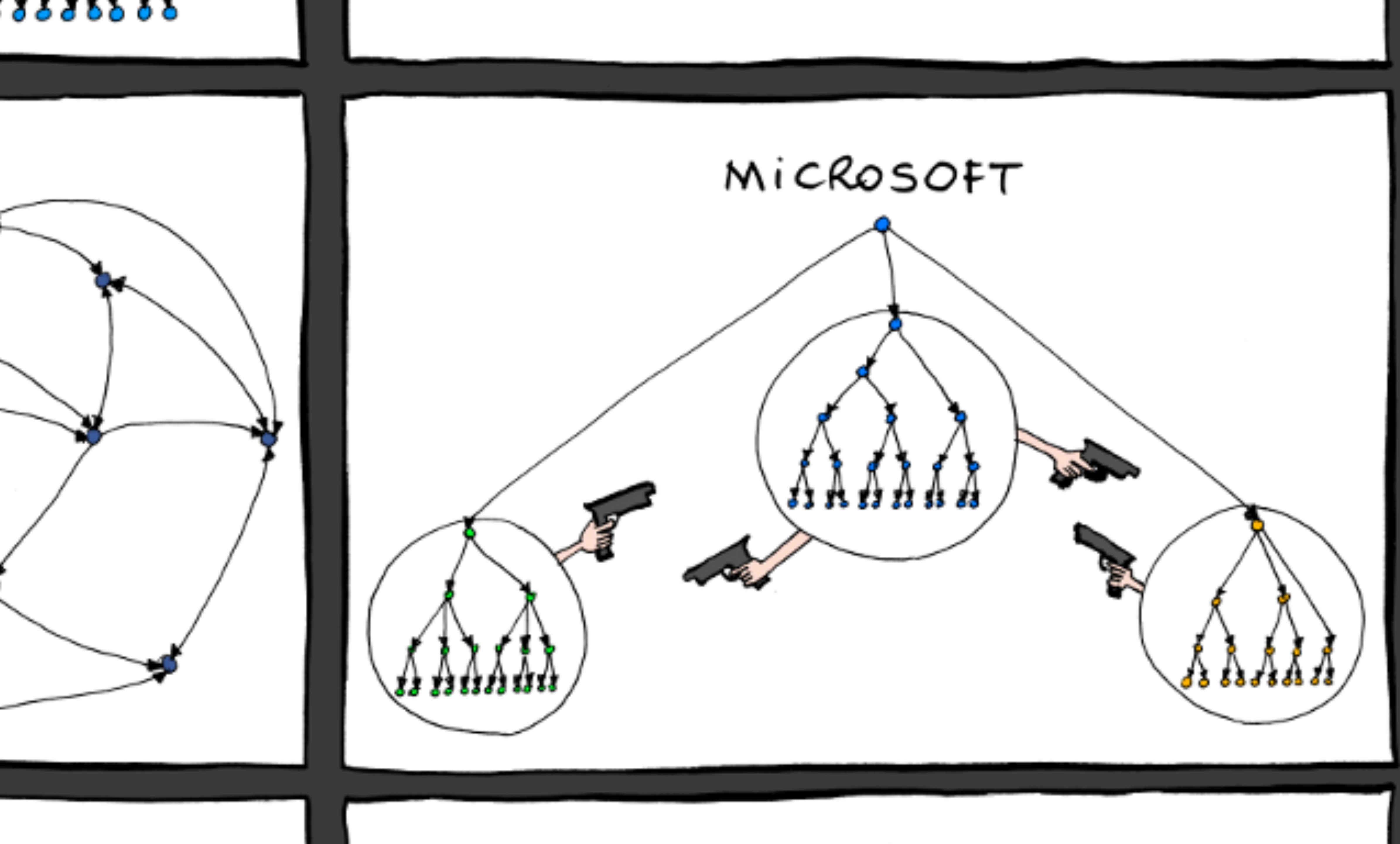
Continuous Improvement



Why transformation?



http://www.bonkersworld.net/images/2011.06.27_organizational_charts.png



http://www.bonkersworld.net/images/2011.06.27_organizational_charts.png

1ES using Azure and GitHub

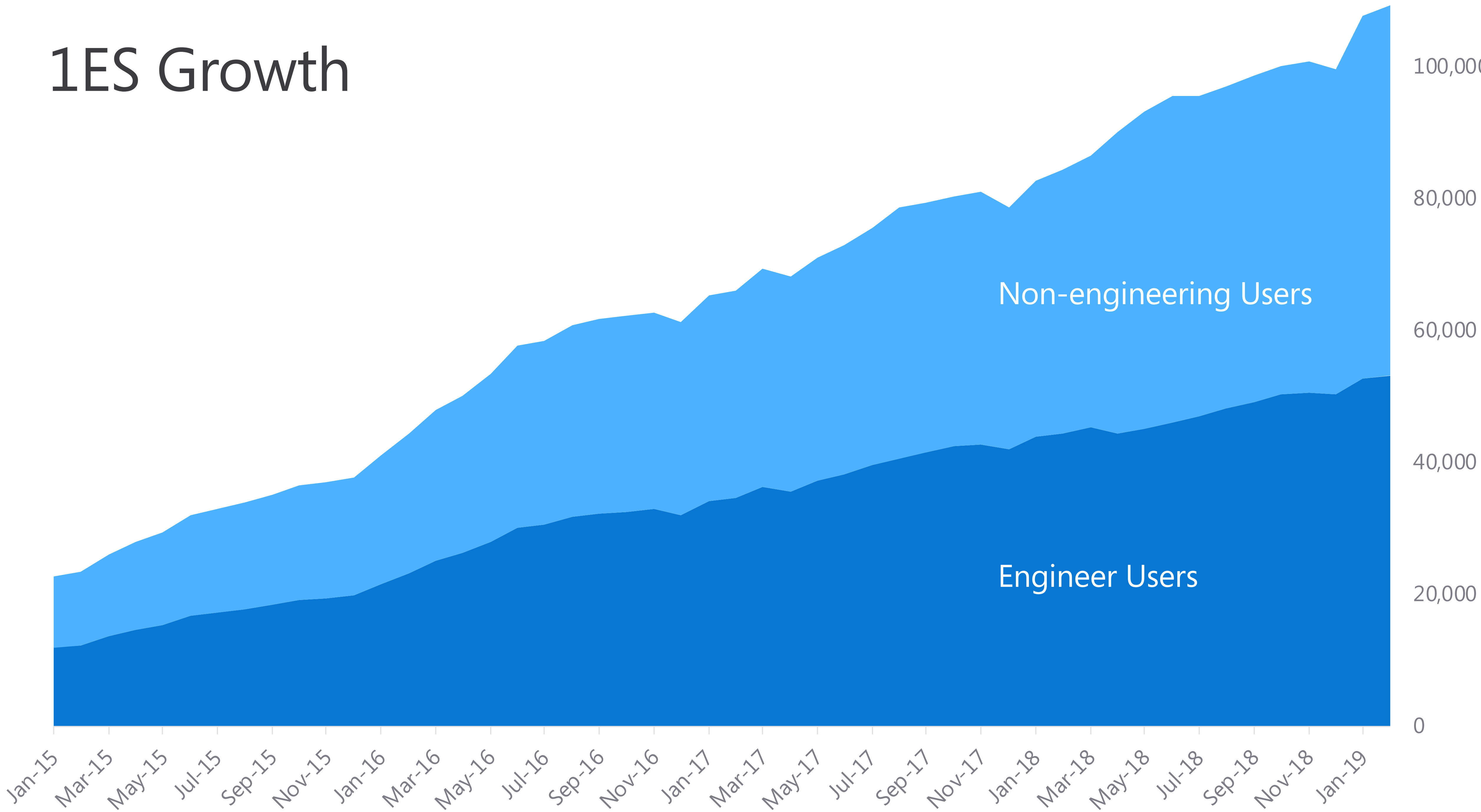
There cannot be a more important thing for an engineer, for a product team, than to work on the systems that drive our productivity.

So I would, any day of the week, trade off features for our own productivity.

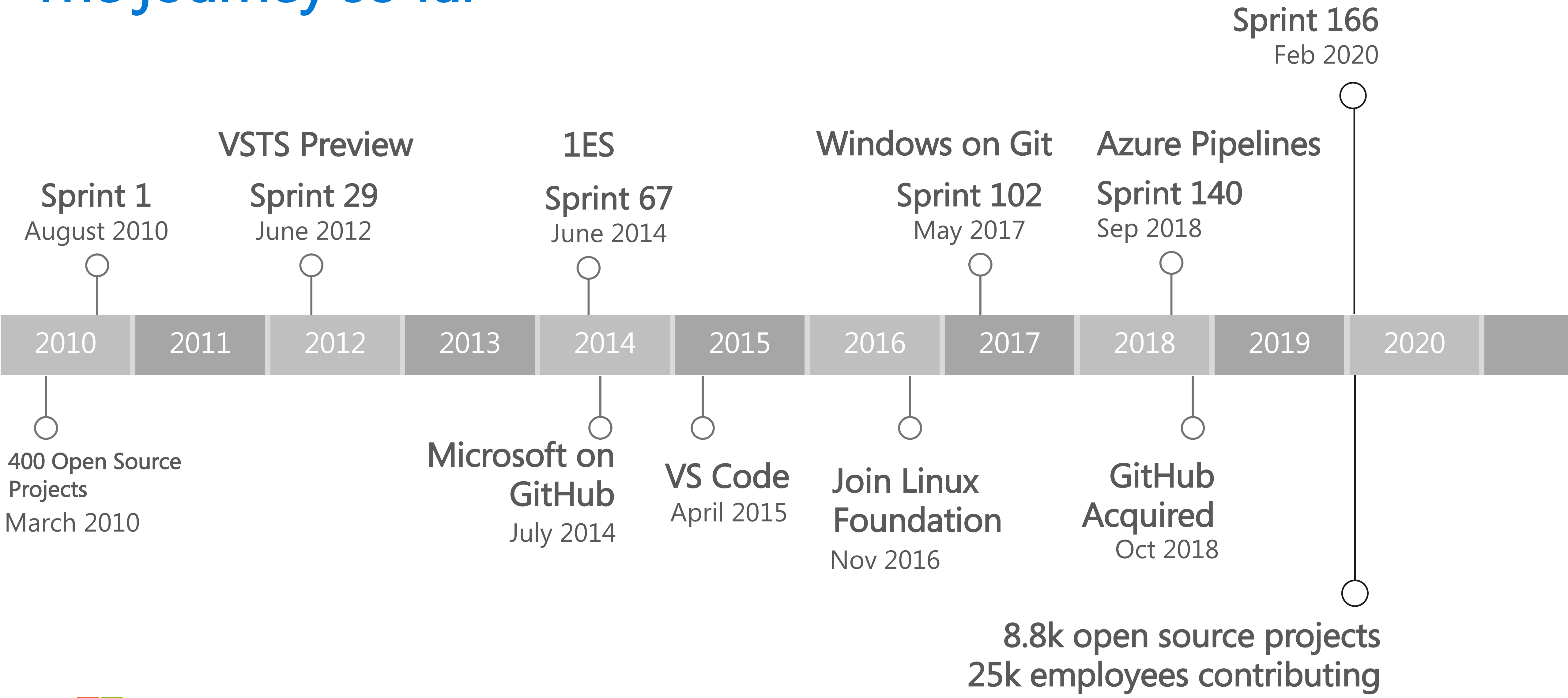
I want our best engineers to work on our engineering systems, so that we can later on come back and build all of the new concepts we want.

- Satya Nadella

1ES Growth

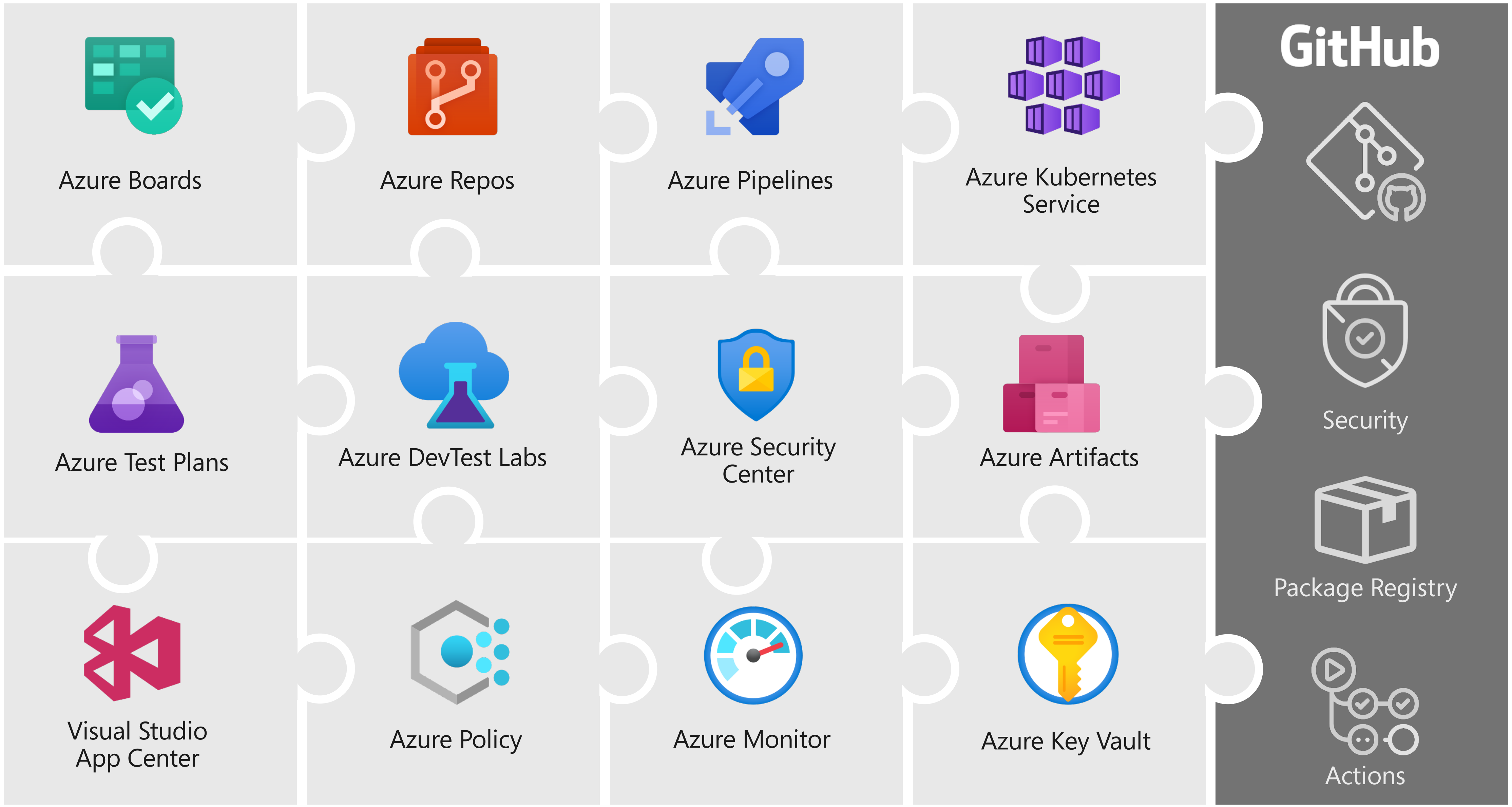

Non-engineering Users
Engineer Users
0
20,000
40,000
60,000
80,000
100,000
Jan-15
Mar-15
May-15
Jul-15
Sep-15
Nov-15
Jan-16
Mar-16
May-16
Jul-16
Sep-16
Nov-16
Jan-17
Mar-17
May-17
Jul-17
Sep-17
Nov-17
Jan-18
Mar-18
May-18
Jul-18
Sep-18
Nov-18
Jan-19

The journey so far



Microsoft

Microsoft's DevOps Tooling – enhanced by GitHub



DevOps at Microsoft

https://aka.ms/DevOpsAtMicrosoft

110k
Active users inside Microsoft

4.6m
Builds per month

28k
Work items created per day

2.4m
Private Git commits per month

8.8k
Open Source repos on GItHub

25k
Employees contributing to open source

82,000
Deployments per day

Data: Internal Microsoft engineering system activity, July 2019

107,000

Engineers in Microsoft working with 1ES

One second per day is like adding

3.7 More people to Microsoft

One minute per day is like adding

163 More people to Microsoft

One hour per day is like adding

$2.7B per year

Microsoft

What did we learn?

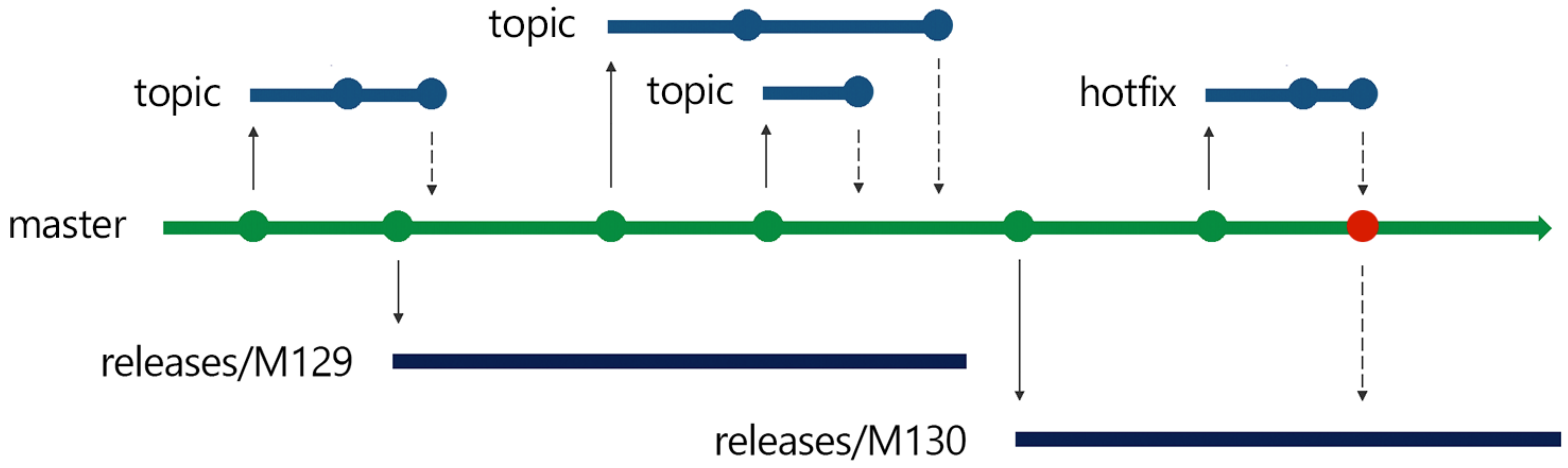
Shipping is a feature!

Our Definition of Done

Live in production, collecting telemetry supporting or diminishing the starting hypothesis.

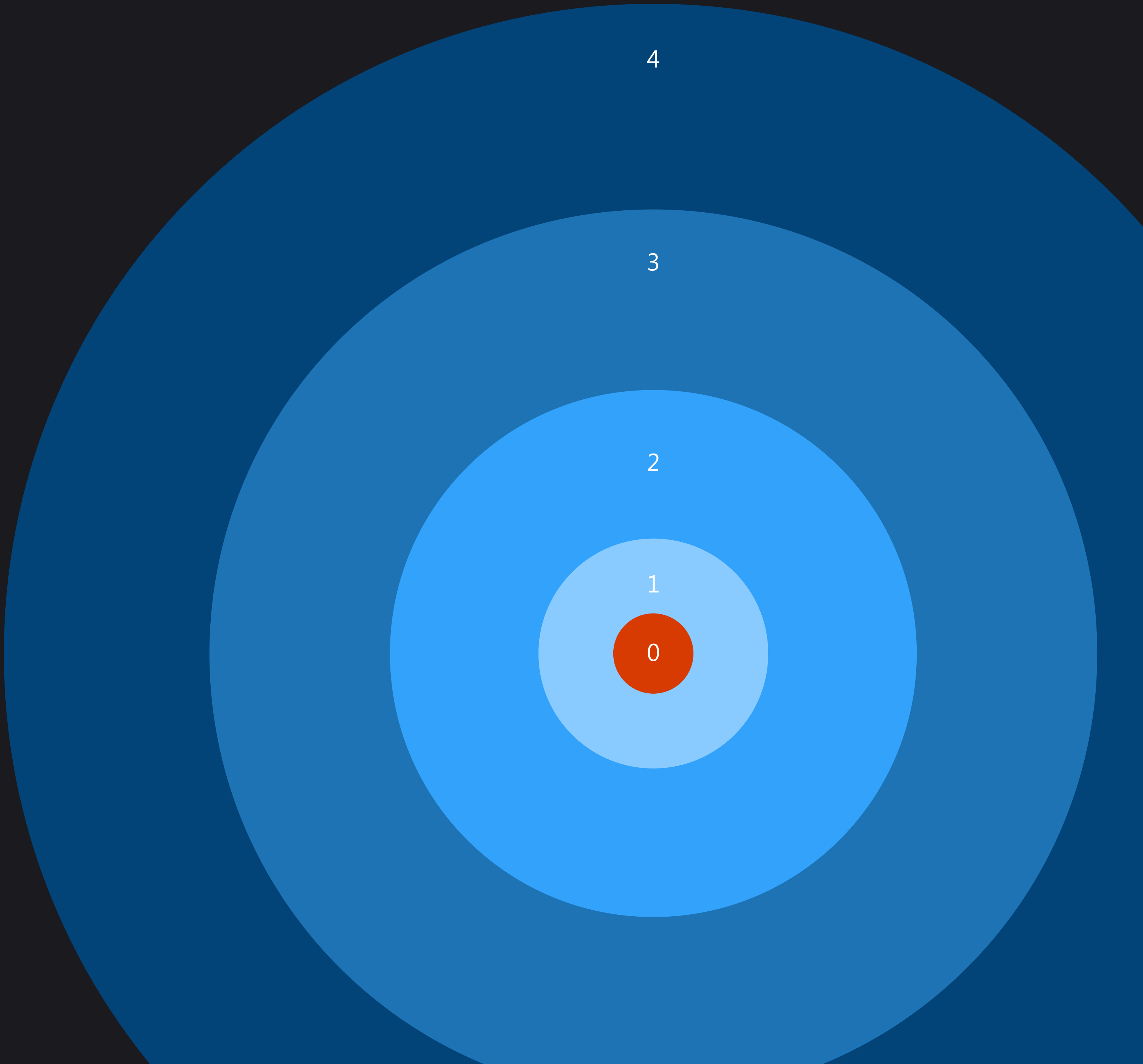
Release Flow

Using Trunk Based Development to avoid Merge Hell

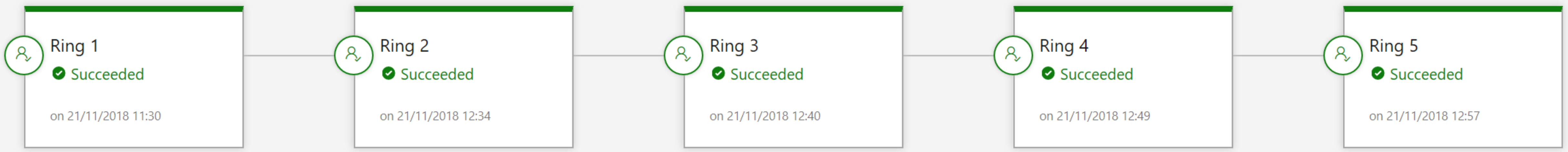


Your aim won’t be perfect.

Control the blast radius.



Tracking Deployments to Production (5 Rings)


Ring 1
Succeeded
on 21/11/2018 11:30

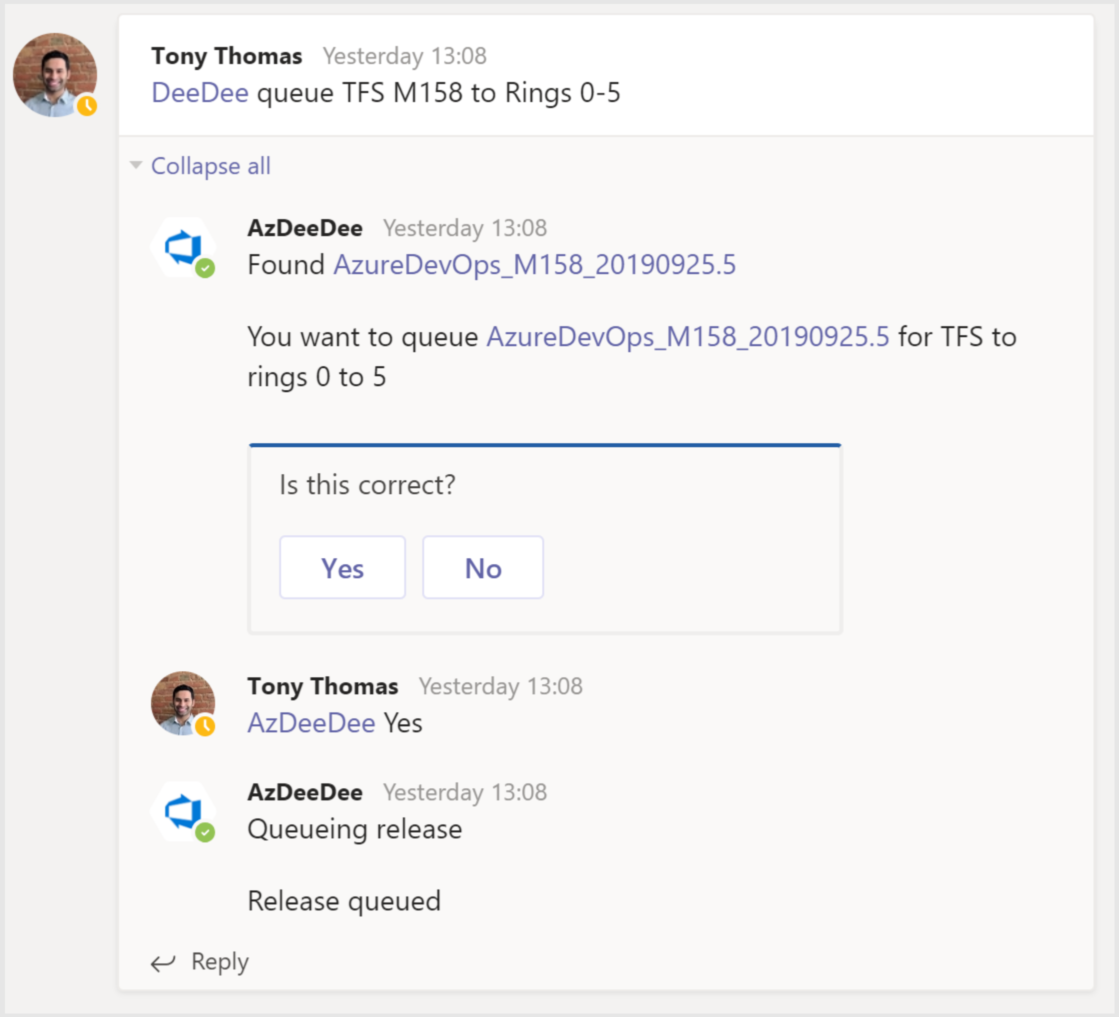
Ring 2
Succeeded
on 21/11/2018 12:34

Ring 3
Succeeded
on 21/11/2018 12:40

Ring 4
Succeeded
on 21/11/2018 12:49

Ring 5
Succeeded
on 21/11/2018 12:57

1. Canary (internal users)
2. Smallest external data center
3. Largest external data center
4. International data centers
5. Everyone


Tony Thomas Yesterday 13:08
DeeDee queue TFS M158 to Rings 0-5

Collapse all

AzDeeDee Yesterday 13:08
Found AzureDevOps_M158_20190925.5

You want to queue AzureDevOps_M158_20190925.5 for TFS to rings 0 to 5

Is this correct?

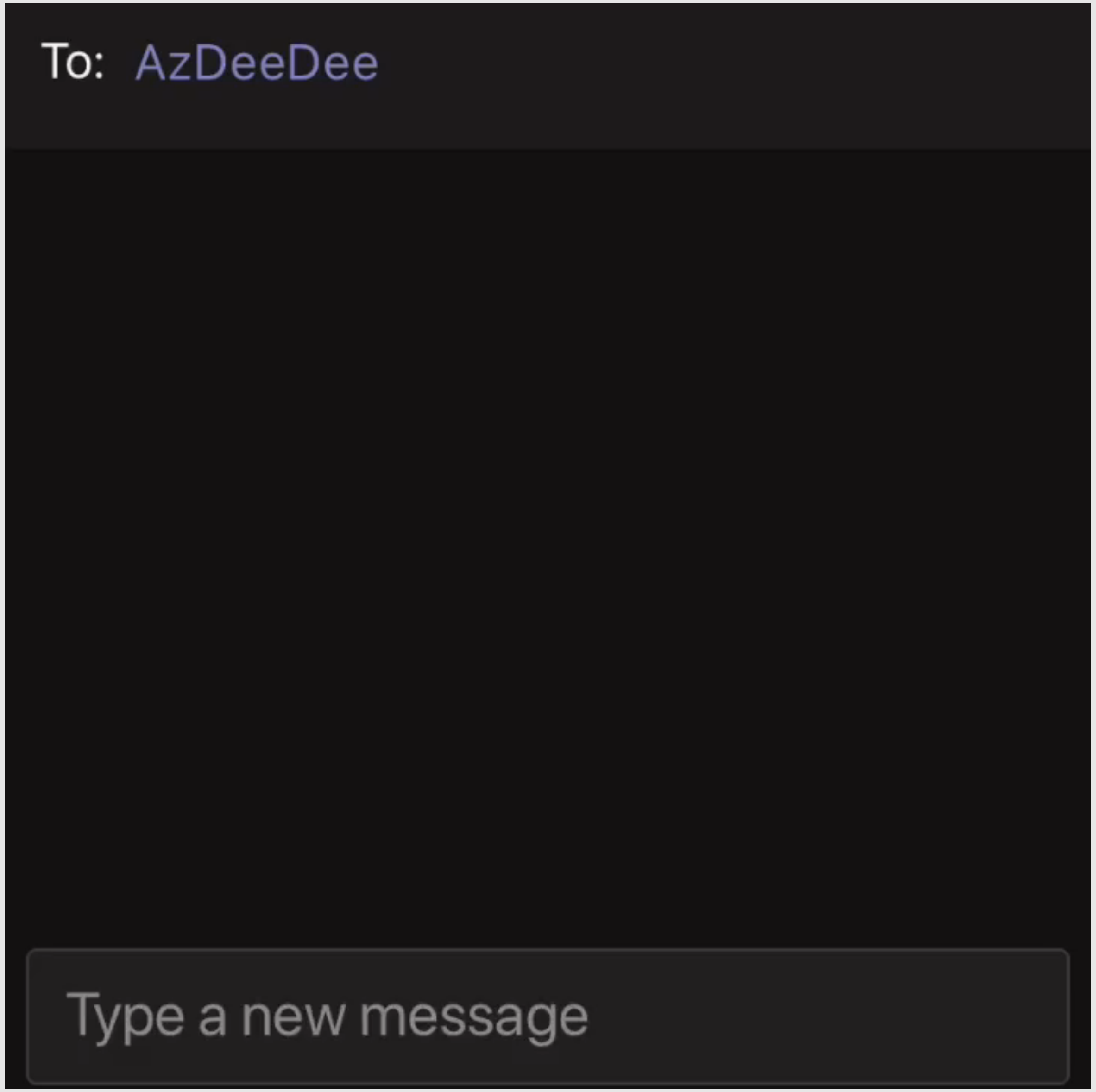
Yes No

Tony Thomas Yesterday 13:08
AzDeeDee Yes

AzDeeDee Yesterday 13:08
Queueing release

Release queued

Reply


To: AzDeeDee

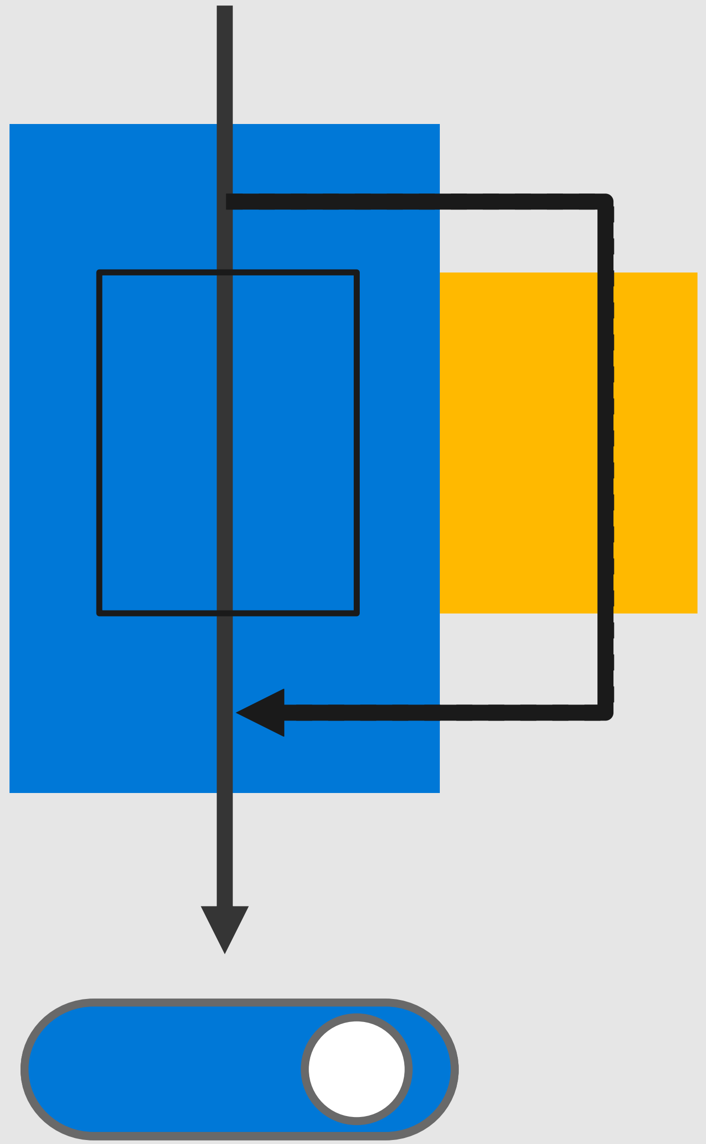
Type a new message

Do you test in production?

Everyone tests in production!

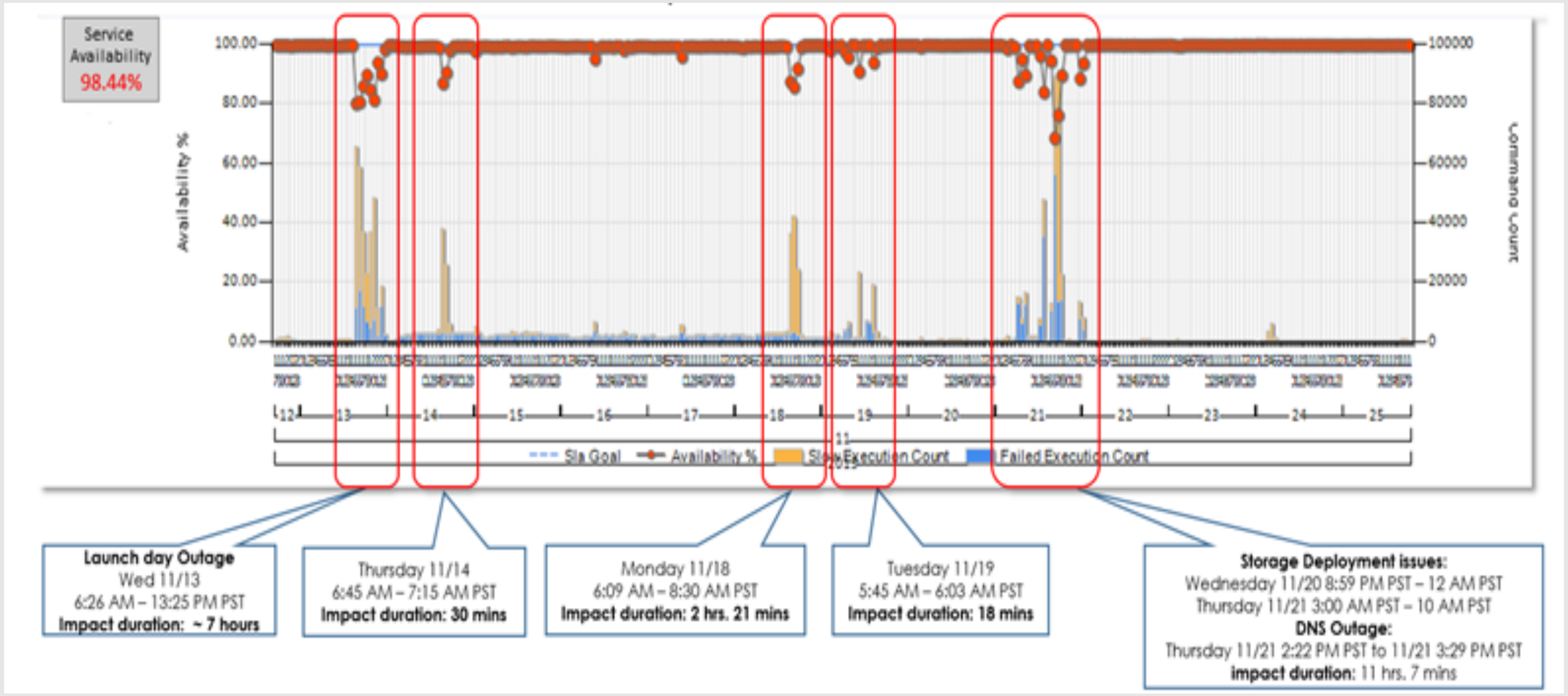
Feature Flags

- All code is deployed
- Runtime control down to individual user
- Enables dark launch



Awesome! What could go wrong?

- We turned features on globally just before the keynote…



Launch day Outage
Wed 11/13
6:26 AM – 13:25 PM PST
Impact duration: ~ 7 hours

Thursday 11/14
6:45 AM – 7:15 AM PST
Impact duration: 30 mins

Monday 11/18
6:09 AM – 8:30 AM PST
Impact duration: 2 hrs. 21 mins

Tuesday 11/19
5:45 AM – 6:03 AM PST
Impact duration: 18 mins

Storage Deployment issues:
Wednesday 11/20 8:59 PM PST – 12 AM PST
Thursday 11/21 3:00 AM PST – 10 AM PST
DNS Outage:
Thursday 11/21 2:22 PM PST to 11/21 3:29 PM PST
impact duration: 11 hrs. 7 mins

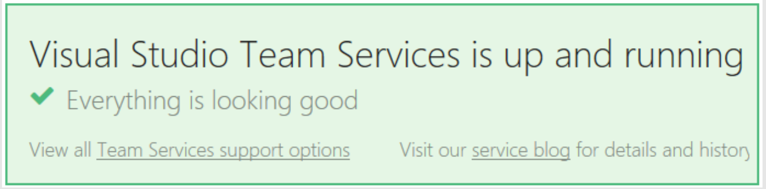
Live Site Culture

Live site status is always the top priority

Live Site Culture

- Live site status is always the top priority
- Weekly live site review
- Root cause everything
- LSI fixes go into backlog (2 sprint rule)
- Actionable alerts
- Monthly service review
- On-call Designated Responsible Individual (DRI)
- Customer Focused Availability model (SLA)
- Per team / service health reports

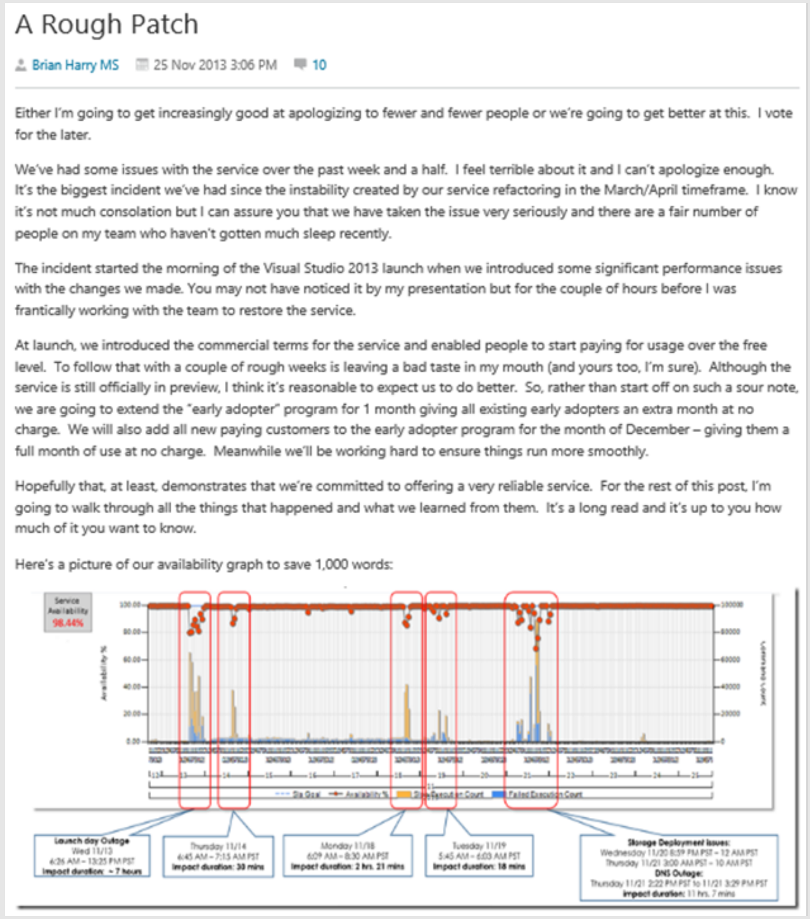
Be Transparent



Visual Studio Team Services is up and running

✓ Everything is looking good

View all Team Services support options Visit our service blog for details and history



A Rough Patch

Brian Harry MS 25 Nov 2013 3:06 PM 10

Either I'm going to get increasingly good at apologizing to fewer and fewer people or we're going to get better at this. I vote for the later.

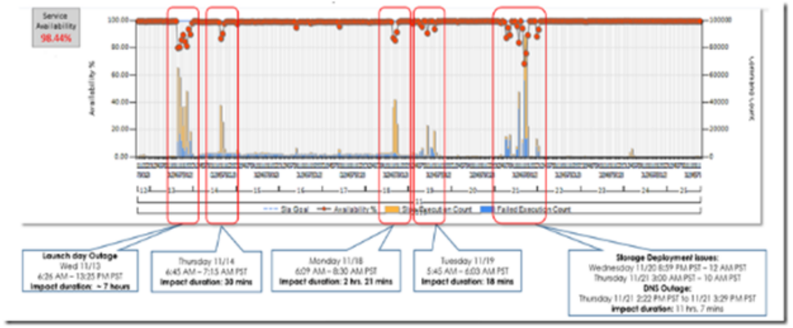
We've had some issues with the service over the past week and a half. I feel terrible about it and I can't apologize enough. It's the biggest incident we've had since the instability created by our service refactoring in the March/April timeframe. I know it's not much consolation but I can assure you that we have taken the issue very seriously and there are a fair number of people on my team who haven't gotten much sleep recently.

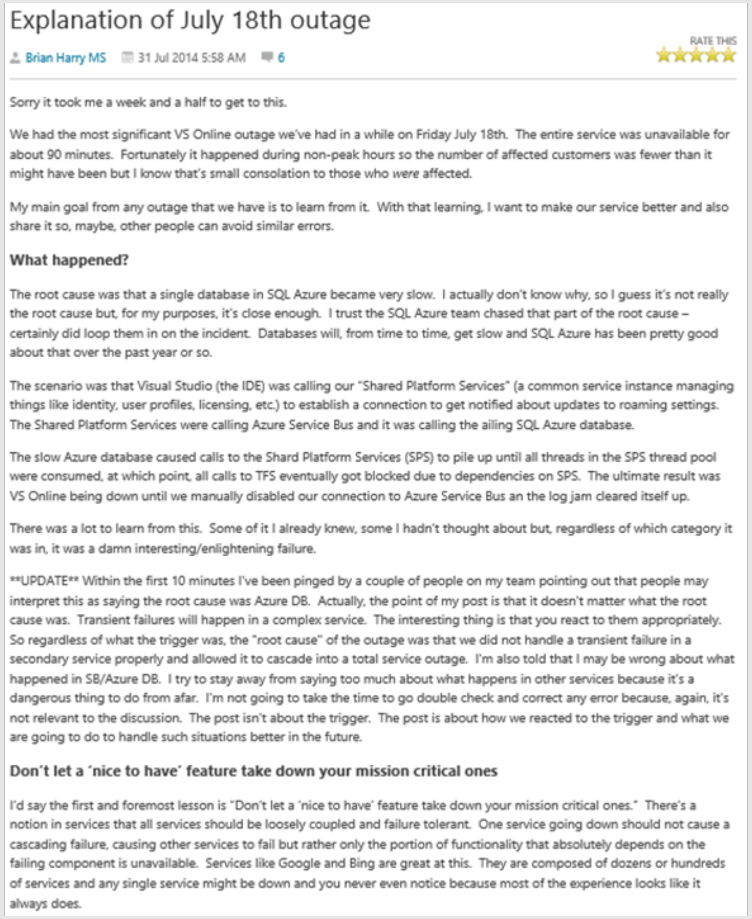
The incident started the morning of the Visual Studio 2013 launch when we introduced some significant performance issues with the changes we made. You may not have noticed it by my presentation but for the couple of hours before I was frantically working with the team to restore the service.

At launch, we introduced the commercial terms for the service and enabled people to start paying for usage over the free level. To follow that with a couple of rough weeks is leaving a bad taste in my mouth (and yours too, I'm sure). Although the service is still officially in preview, I think it's reasonable to expect us to do better. So, rather than start off on such a sour note, we are going to extend the "early adopter" program for 1 month giving all existing early adopters an extra month at no charge. We will also add all new paying customers to the early adopter program for the month of December – giving them a full month of use at no charge. Meanwhile we'll be working hard to ensure things run more smoothly.

Hopefully that, at least, demonstrates that we're committed to offering a very reliable service. For the rest of this post, I'm going to walk through all the things that happened and what we learned from them. It's a long read and it's up to you how much of it you want to know.

Here's a picture of our availability graph to save 1,000 words:





Explanation of July 18th outage

Brian Harry MS 31 Jul 2014 5:58 AM 6 RATE THIS ★★★★★

Sorry it took me a week and a half to get to this.

We had the most significant VS Online outage we've had in a while on Friday July 18th. The entire service was unavailable for about 90 minutes. Fortunately it happened during non-peak hours so the number of affected customers was fewer than it might have been but I know that's small consolation to those who *were* affected.

My main goal from any outage that we have is to learn from it. With that learning, I want to make our service better and also share it so, maybe, other people can avoid similar errors.

What happened?

The root cause was that a single database in SQL Azure became very slow. I actually don't know why, so I guess it's not really the root cause but, for my purposes, it's close enough. I trust the SQL Azure team chased that part of the root cause – certainly did loop them in on the incident. Databases will, from time to time, get slow and SQL Azure has been pretty good about that over the past year or so.

The scenario was that Visual Studio (the IDE) was calling our "Shared Platform Services" (a common service instance managing things like identity, user profiles, licensing, etc.) to establish a connection to get notified about updates to roaming settings. The Shared Platform Services were calling Azure Service Bus and it was calling the ailing SQL Azure database.

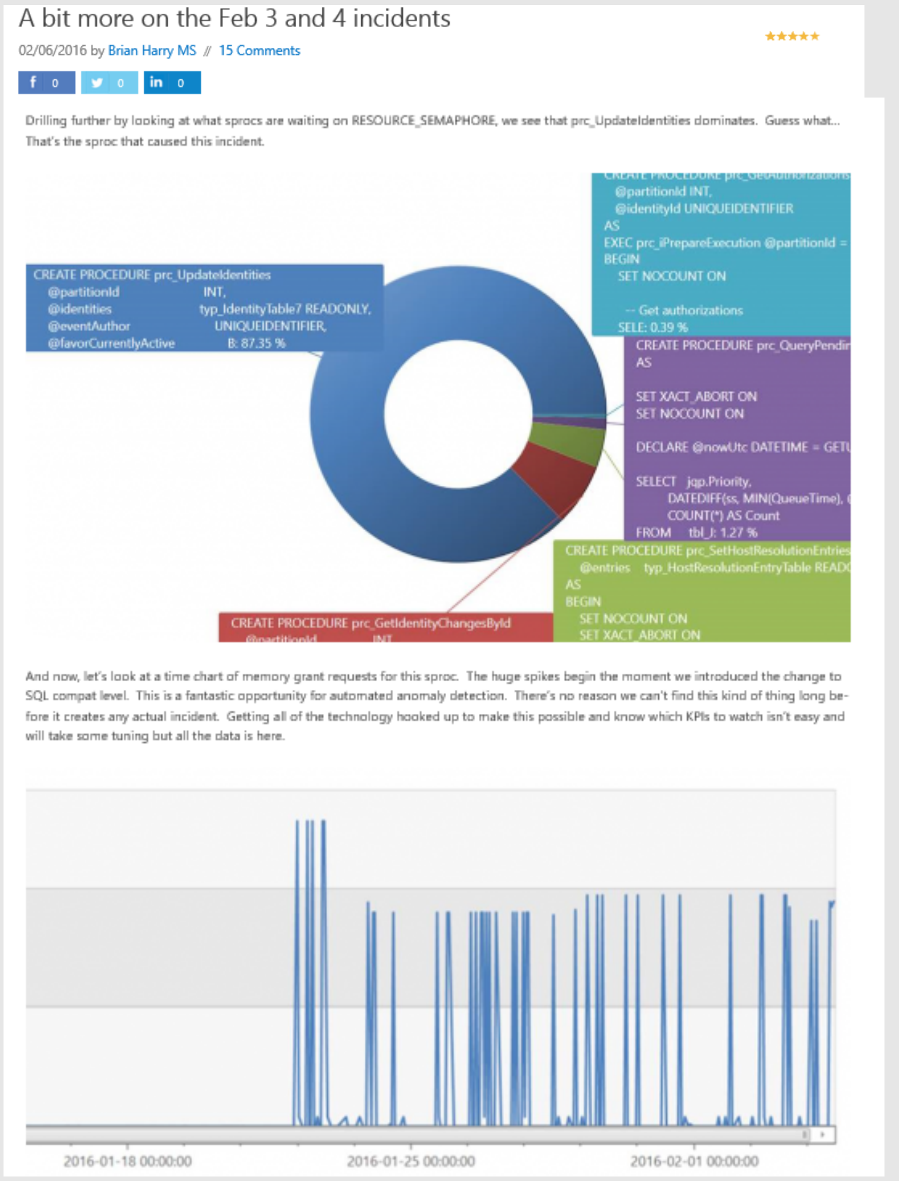
The slow Azure database caused calls to the Shard Platform Services (SPS) to pile up until all threads in the SPS thread pool were consumed, at which point, all calls to TFS eventually got blocked due to dependencies on SPS. The ultimate result was VS Online being down until we manually disabled our connection to Azure Service Bus an the log jam cleared itself up.

There was a lot to learn from this. Some of it I already knew, some I hadn't thought about but, regardless of which category it was in, it was a damn interesting/enlightening failure.

UPDATE Within the first 10 minutes I've been pinged by a couple of people on my team pointing out that people may interpret this as saying the root cause was Azure DB. Actually, the point of my post is that it doesn't matter what the root cause was. Transient failures will happen in a complex service. The interesting thing is that you react to them appropriately. So regardless of what the trigger was, the "root cause" of the outage was that we did not handle a transient failure in a secondary service properly and allowed it to cascade into a total service outage. I'm also told that I may be wrong about what happened in SB/Azure DB. I try to stay away from saying too much about what happens in other services because it's a dangerous thing to do from afar. I'm not going to take the time to go double check and correct any error because, again, it's not relevant to the discussion. The post isn't about the trigger. The post is about how we reacted to the trigger and what we are going to do to handle such situations better in the future.

Don't let a 'nice to have' feature take down your mission critical ones

I'd say the first and foremost lesson is "Don't let a 'nice to have' feature take down your mission critical ones." There's a notion in services that all services should be loosely coupled and failure tolerant. One service going down should not cause a cascading failure, causing other services to fail but rather only the portion of functionality that absolutely depends on the failing component is unavailable. Services like Google and Bing are great at this. They are composed of dozens or hundreds of services and any single service might be down and you never even notice because most of the experience looks like it always does.

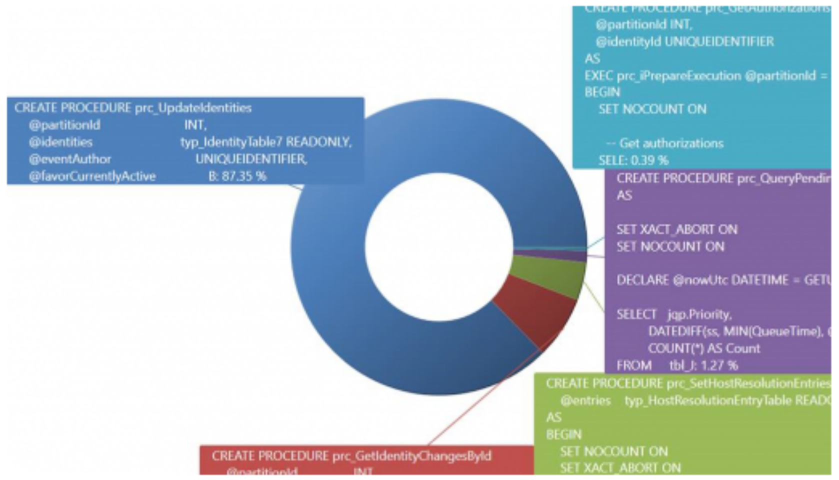


A bit more on the Feb 3 and 4 incidents

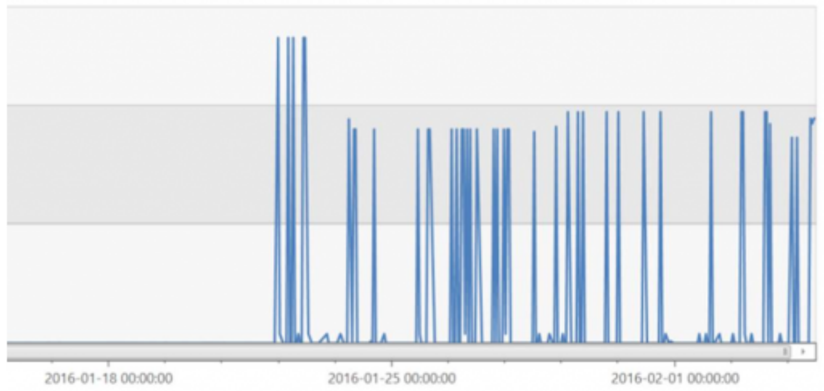
02/06/2016 by Brian Harry MS // 15 Comments ★★★★★

f 0 0 in 0

Drilling further by looking at what sprocs are waiting on RESOURCE_SEMAPHORE, we see that prc_UpdateIdentities dominates. Guess what... That's the sproc that caused this incident.



And now, let's look at a time chart of memory grant requests for this sproc. The huge spikes begin the moment we introduced the change to SQL compat level. This is a fantastic opportunity for automated anomaly detection. There's no reason we can't find this kind of thing long before it creates any actual incident. Getting all of the technology hooked up to make this possible and know which KPIs to watch isn't easy and will take some tuning but all the data is here.



Iterate over pain

No such thing as ‘partial automation’

“One time” deployment commands in OneNote, email

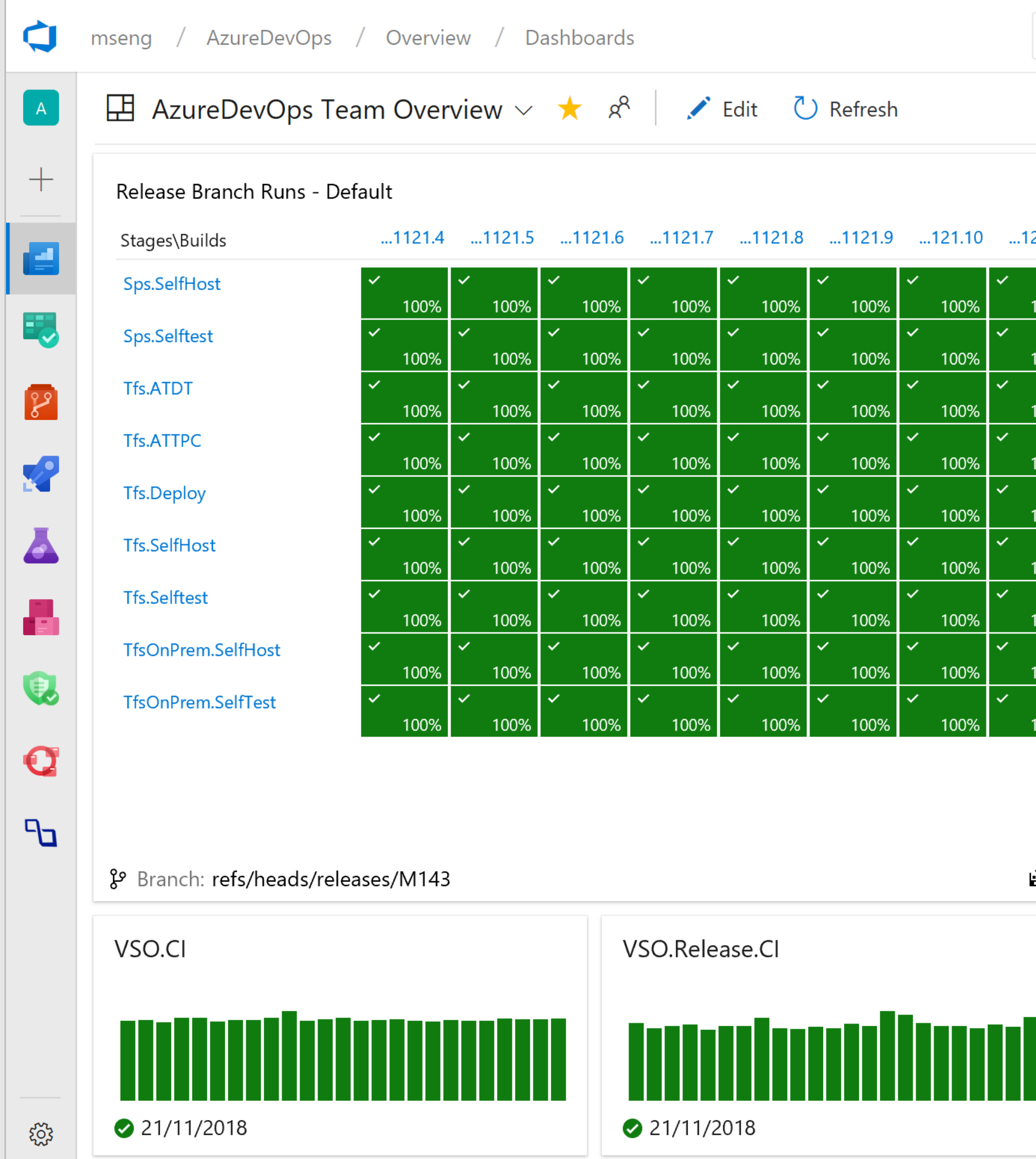
```
Set-Options “-p 0”
```

Imagine a dozen more steps like that…

And then…someone misses a step halfway through

Automate completely

- No more “one time” commands run manually
- Pre-production & canary are the same as production **every time**


mseng / AzureDevOps / Overview / Dashboards
AzureDevOps Team Overview
Edit
Refresh

Release Branch Runs - Default

Stages\Builds	...1121.4	...1121.5	...1121.6	...1121.7	...1121.8	...1121.9	...121.10	...12
Sps.SelfHost	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 1
Sps.Selftest	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 1
Tfs.ATDT	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 1
Tfs.ATTPC	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 1
Tfs.Deploy	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 1
Tfs.SelfHost	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 1
Tfs.Selftest	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 1
TfsOnPrem.SelfHost	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 1
TfsOnPrem.SelfTest	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 1

Branch: refs/heads/releases/M143

VSO.CI

✓ 21/11/2018

VSO.Release.CI

✓ 21/11/2018

If a process often breaks,

Do it MORE often

Shift left quality

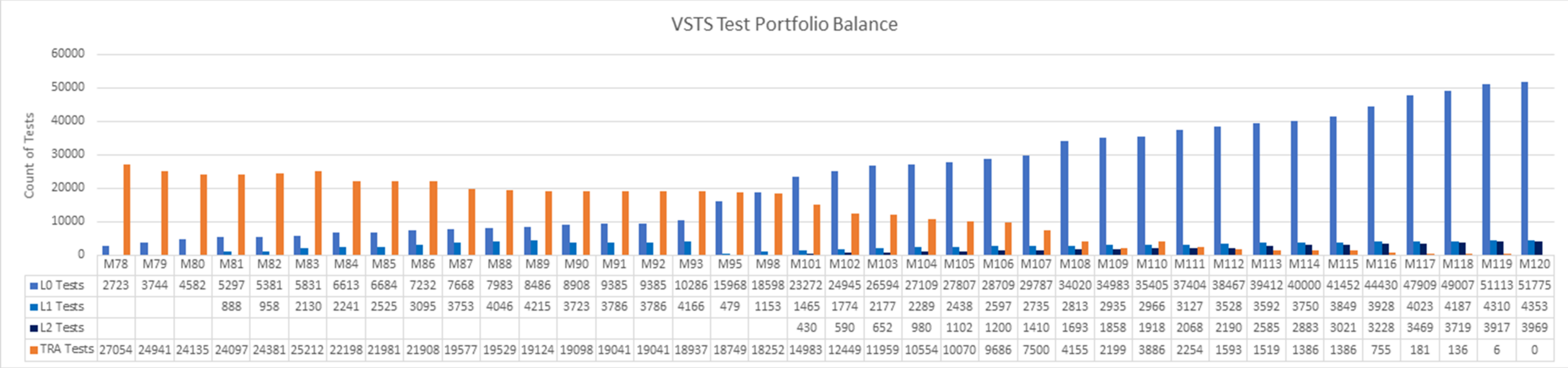
Testing: Shift Left from Integration to Unit

L0 – Requires only built binaries, no dependencies

L1 – Adds ability to use SQL and file system
  Run L0 & L1 in the pull request builds

L2 – Test a service via REST APIs

L3 – Full environment to test end to end



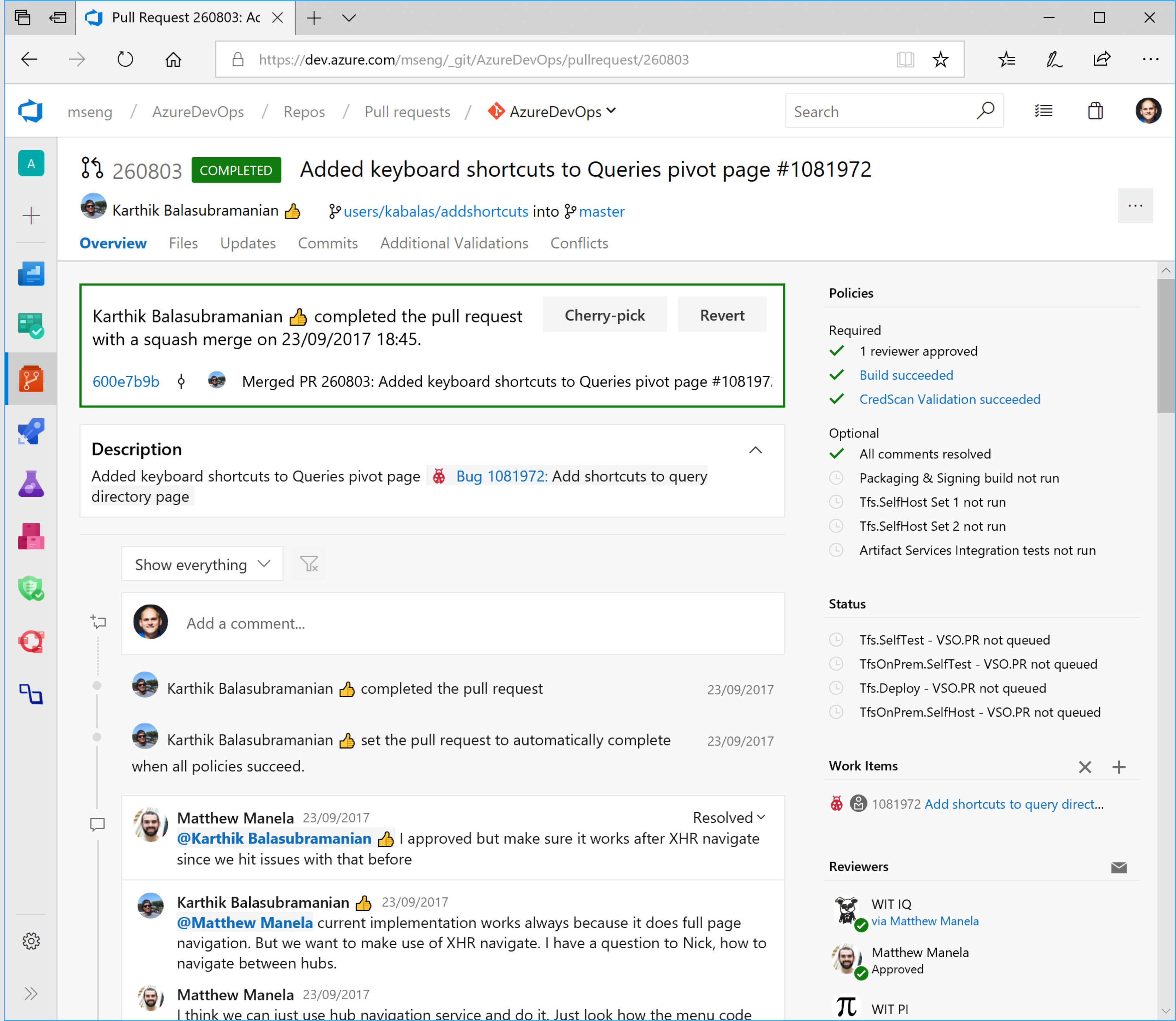
	M78	M79	M80	M81	M82	M83	M84	M85	M86	M87	M88	M89	M90	M91	M92	M93	M95	M98
■ L0 Tests	2723	3744	4582	5297	5381	5831	6613	6684	7232	7668	7983	8486	8908	9385	9385	10286	15968	18598
■ L1 Tests				888	958	2130	2241	2525	3095	3753	4046	4215	3723	3786	3786	4166	479	1153
■ L2 Tests																		
■ TRA Tests	27054	24941	24135	24097	24381	25212	22198	21981	21908	19577	19529	19124	19098	19041	19041	18937	18749	18252

	M101	M102	M103	M104	M105	M106	M107	M108	M109	M110	M111	M112	M113	M114	M115	M116	M117	M118	M119	M120
■ L0 Tests	23272	24945	26594	27109	27807	28709	29787	34020	34983	35405	37404	38467	39412	40000	41452	44430	47909	49007	51113	51775
■ L1 Tests	1465	1774	2177	2289	2438	2597	2735	2813	2935	2966	3127	3528	3592	3750	3849	3928	4023	4187	4310	4353
■ L2 Tests	430	590	652	980	1102	1200	1410	1693	1858	1918	2068	2190	2585	2883	3021	3228	3469	3719	3917	3969
■ TRA Tests	14983	12449	11959	10554	10070	9686	7500	4155	2199	3886	2254	1593	1519	1386	1386	755	181	136	6	0

Pull Requests

PR's are point of code review

L0+L1 Tests performed before merge



Pull Request 260803: Ac

https://dev.azure.com/mseng/_git/AzureDevOps/pullrequest/260803

mseng / AzureDevOps / Repos / Pull requests / AzureDevOps

260803 COMPLETED Added keyboard shortcuts to Queries pivot page #1081972

Karthik Balasubramanian users/kabalas/addshortcuts into master

Overview Files Updates Commits Additional Validations Conflicts

Karthik Balasubramanian completed the pull request with a squash merge on 23/09/2017 18:45. Cherry-pick Revert

600e7b9b Merged PR 260803: Added keyboard shortcuts to Queries pivot page #108197

Description

Added keyboard shortcuts to Queries pivot page Bug 1081972: Add shortcuts to query directory page

Show everything

Add a comment...

Karthik Balasubramanian completed the pull request 23/09/2017

Karthik Balasubramanian set the pull request to automatically complete when all policies succeed. 23/09/2017

Matthew Manela 23/09/2017 Resolved
@Karthik Balasubramanian I approved but make sure it works after XHR navigate since we hit issues with that before

Karthik Balasubramanian 23/09/2017
@Matthew Manela current implementation works always because it does full page navigation. But we want to make use of XHR navigate. I have a question to Nick, how to navigate between hubs.

Matthew Manela 23/09/2017
I think we can just use hub navigation service and do it. Just look how the menu code

Policies

Required
- ✓ 1 reviewer approved
- ✓ Build succeeded
- ✓ CredScan Validation succeeded

Optional
- ✓ All comments resolved
- Packaging & Signing build not run
- Tfs.SelfHost Set 1 not run
- Tfs.SelfHost Set 2 not run
- Artifact Services Integration tests not run

Status
- Tfs.SelfTest - VSO.PR not queued
- TfsOnPrem.SelfTest - VSO.PR not queued
- Tfs.Deploy - VSO.PR not queued
- TfsOnPrem.SelfHost - VSO.PR not queued

Work Items

1081972 Add shortcuts to query direct...

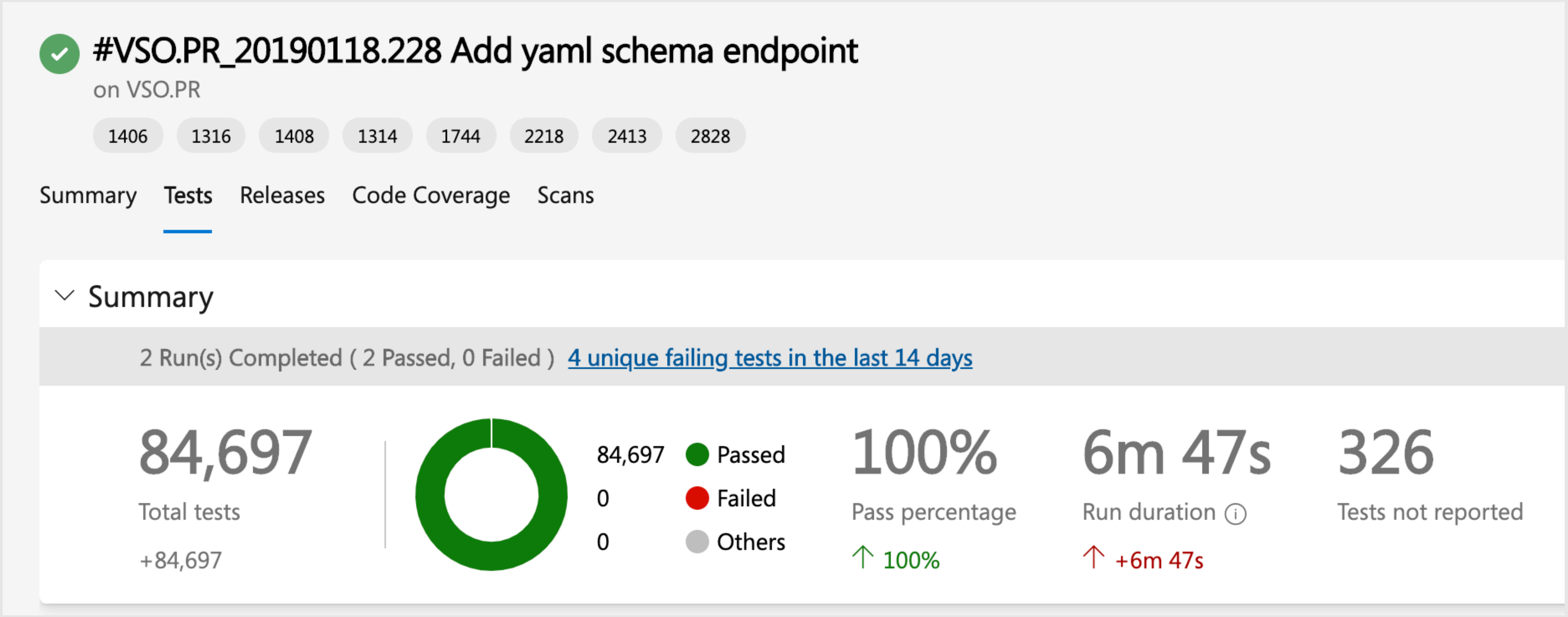
Reviewers

WIT IQ via Matthew Manela

Matthew Manela Approved

WIT PI

Tests Against the Pull Request



#VSO.PR_20190118.228 Add yaml schema endpoint

on VSO.PR

1406 1316 1408 1314 1744 2218 2413 2828

Summary Tests Releases Code Coverage Scans

Summary

2 Run(s) Completed (2 Passed, 0 Failed) 4 unique failing tests in the last 14 days

84,697
Total tests
+84,697

84,697	● Passed
0	● Failed
0	● Others

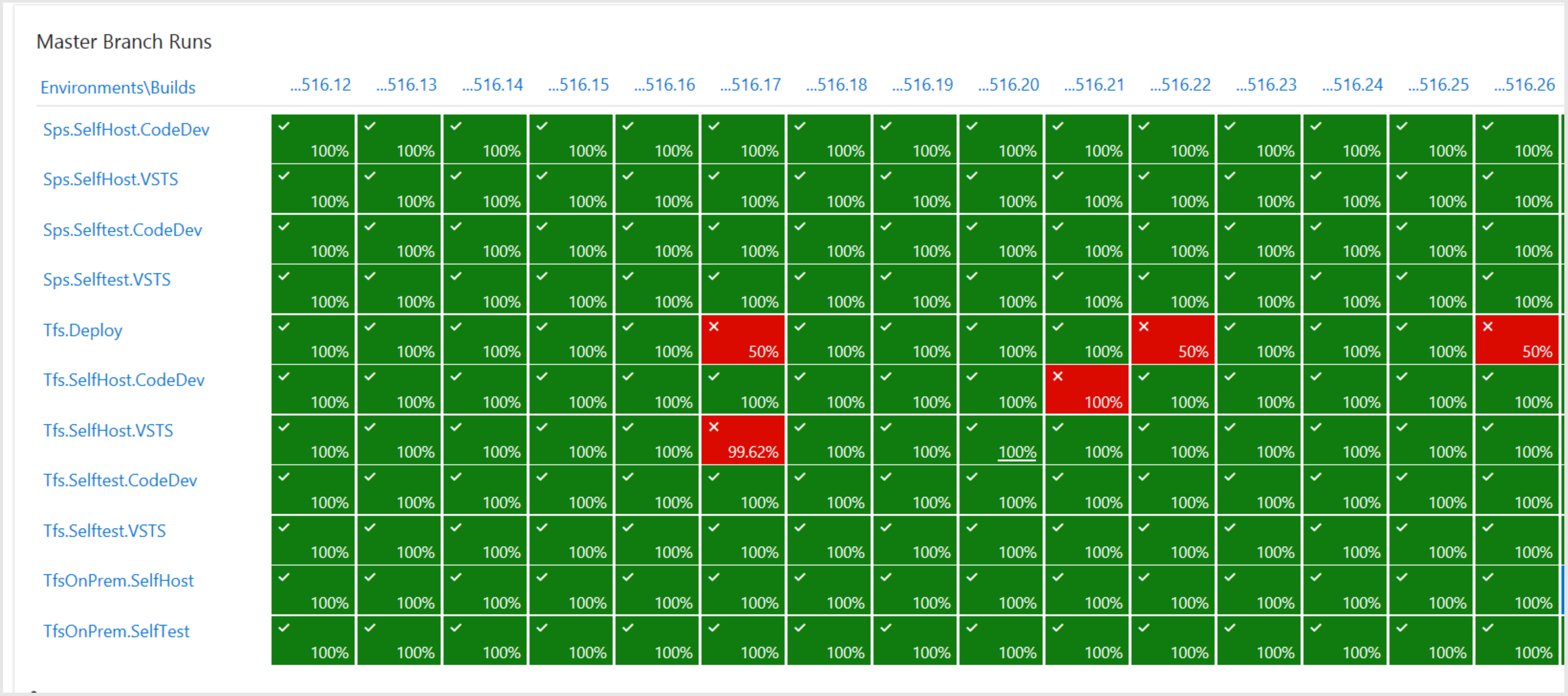
100%
Pass percentage
↑ 100%

6m 47s
Run duration
↑ +6m 47s

326
Tests not reported

Feedback in minutes, before acceptance of PR

Green Means Green, Red Means Red



Master Branch Runs

Environments\Builds	...516.12	...516.13	...516.14	...516.15	...516.16	...516.17	...516.18	...516.19	...516.20	...516.21	...516.22	...516.23	...516.24	...516.25	...516.26
Sps.SelfHost.CodeDev	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%
Sps.SelfHost.VSTS	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%
Sps.Selftest.CodeDev	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%
Sps.Selftest.VSTS	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%
Tfs.Deploy	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✗ 50%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✗ 50%	✓ 100%	✓ 100%	✓ 100%	✗ 50%
Tfs.SelfHost.CodeDev	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✗ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%
Tfs.SelfHost.VSTS	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✗ 99.62%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%
Tfs.Selftest.CodeDev	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%
Tfs.Selftest.VSTS	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%
TfsOnPrem.SelfHost	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%
TfsOnPrem.SelfTest	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%

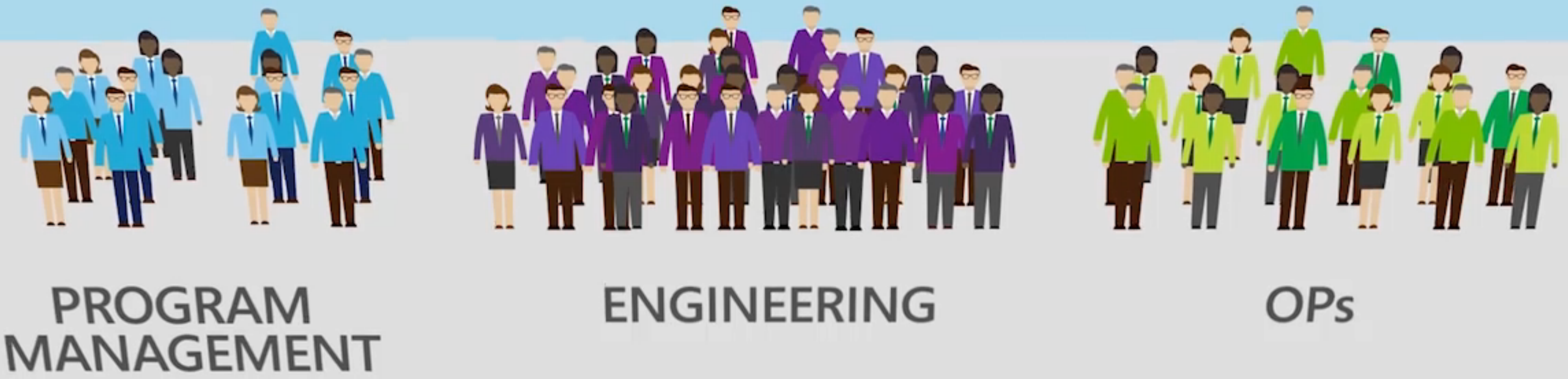
Only all-green builds get to release

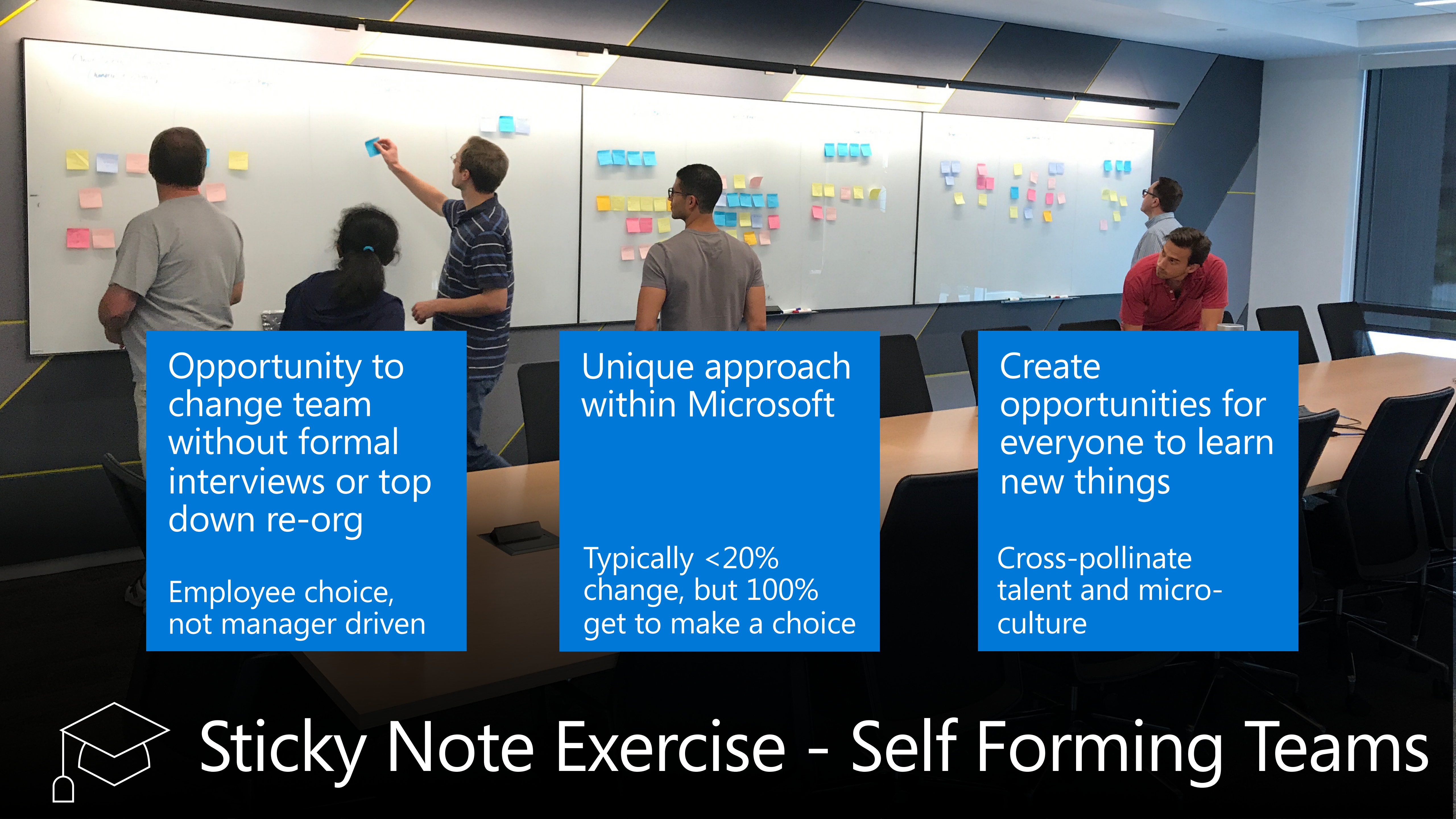
Autonomy vs alignment

Team Structure

ORG CHART

ORG CHART





Opportunity to change team without formal interviews or top down re-org

Employee choice, not manager driven

Unique approach within Microsoft

Typically <20% change, but 100% get to make a choice

Create opportunities for everyone to learn new things

Cross-pollinate talent and micro-culture

Sticky Note Exercise - Self Forming Teams

Measure impact not activity

Measure what's important (KPI's)

Usage

- Acquisition
- Engagement
- Satisfaction
- Churn
- Feature Usage

I2D

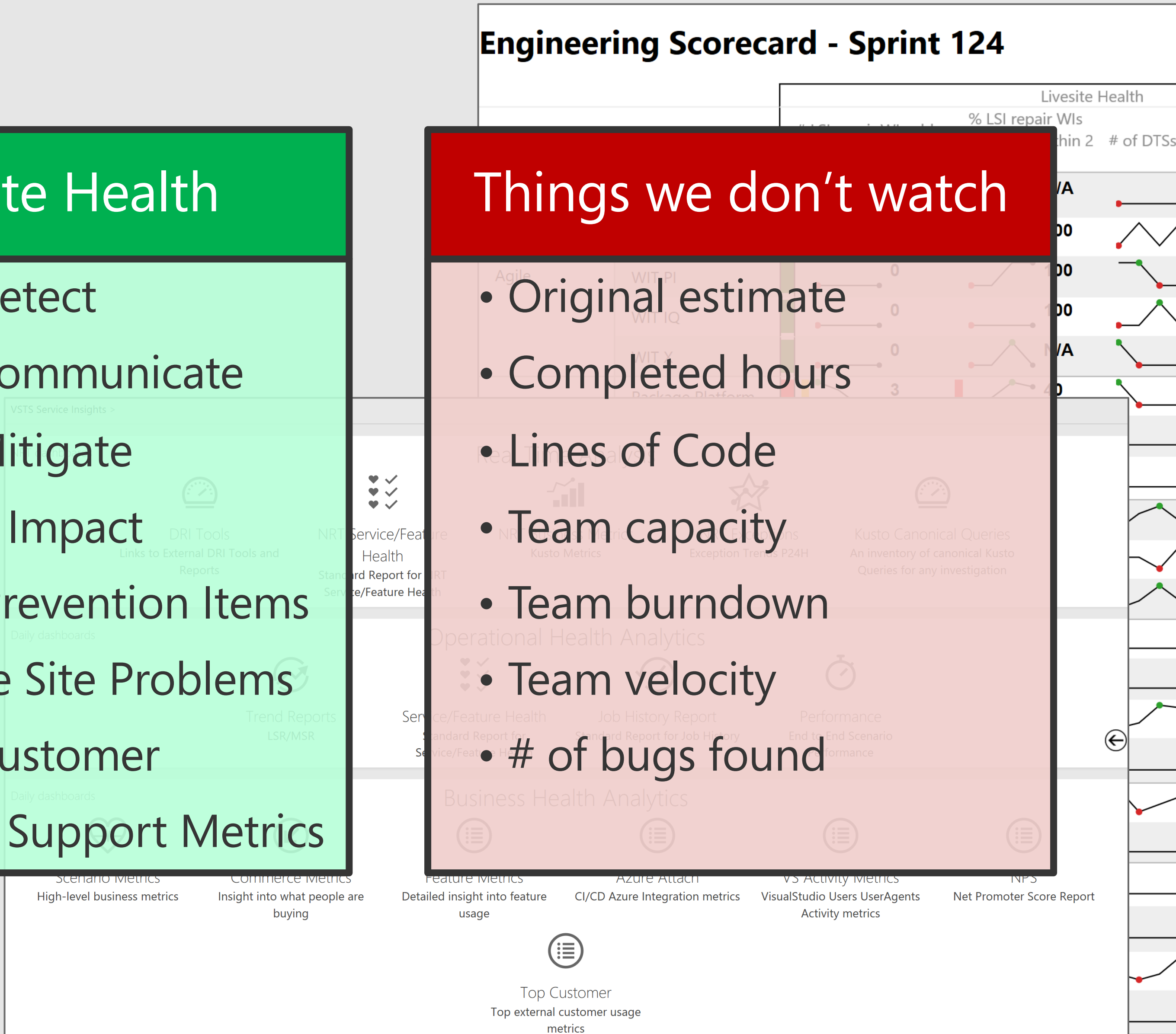
- Time to Build
- Time to Self Test
- Time to Deploy
- Time to Learn

Live Site Health

- Time to Detect
- Time to Communicate
- Time to Mitigate
- Customer Impact
- Incident Prevention Items
- Aging Live Site Problems
- SLA per Customer
- Customer Support Metrics

Things we don't watch

- Original estimate
- Completed hours
- Lines of Code
- Team capacity
- Team burndown
- Team velocity
- # of bugs found



Engineering Scorecard - Sprint 124

Livesite Health

% LSI repair WIs

# of DTSs

Scenario Metrics
High-level business metrics

Commerce Metrics
Insight into what people are buying

Feature Metrics
Detailed insight into feature usage

Azure Attach
CI/CD Azure Integration metrics

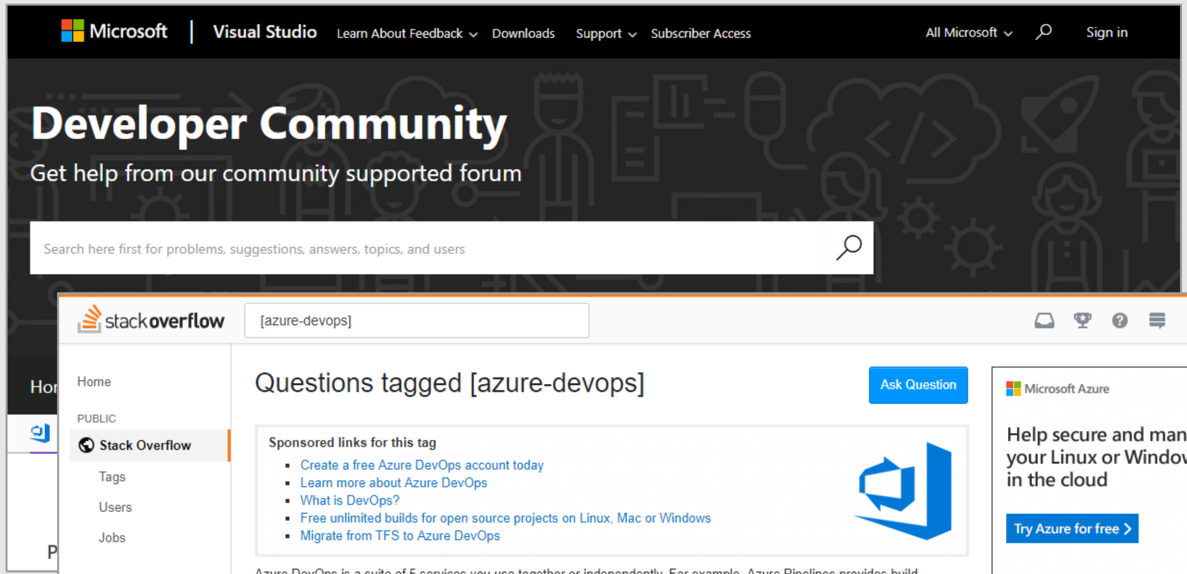
VS Activity Metrics
VisualStudio Users UserAgents Activity metrics

NPS
Net Promoter Score Report

Top Customer
Top external customer usage metrics

Listen to our customers

Quantitively & Qualitatively

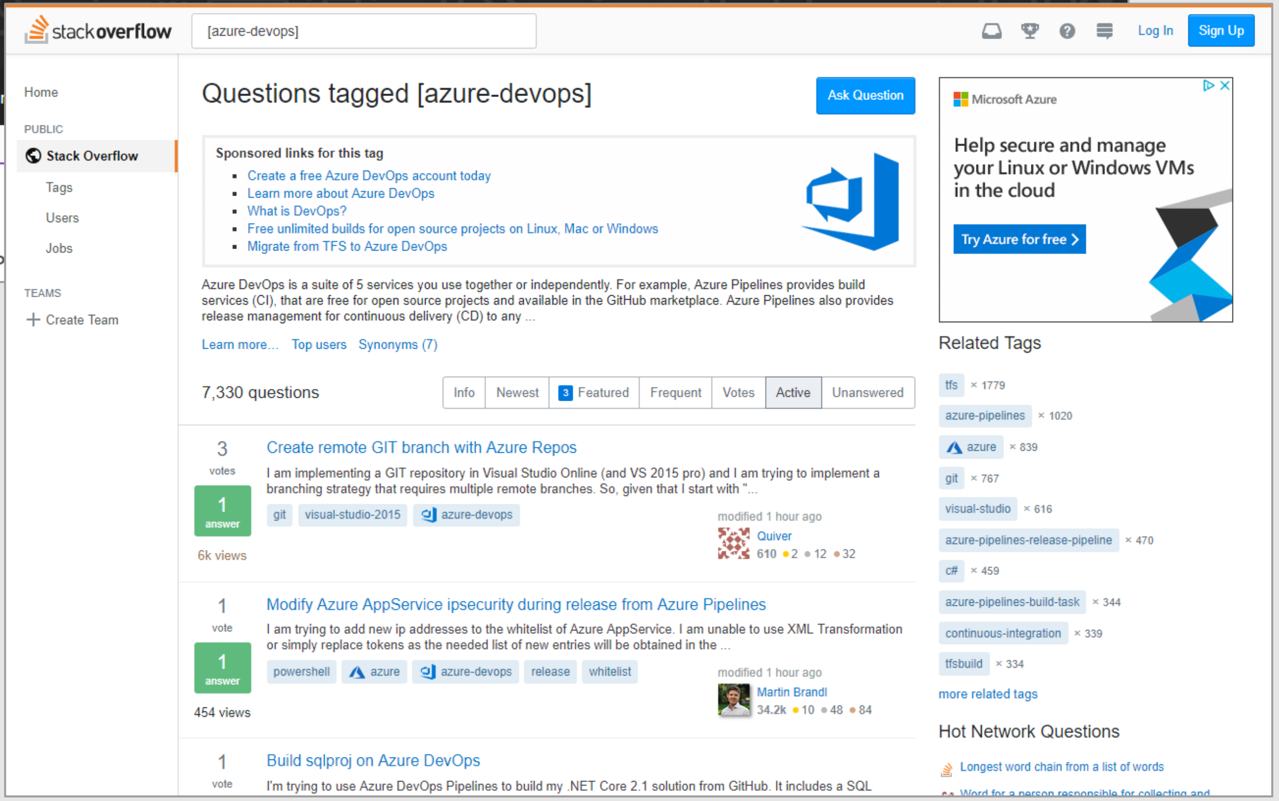


Microsoft | Visual Studio Learn About Feedback Downloads Support Subscriber Access All Microsoft Sign in

Developer Community

Get help from our community supported forum

Search here first for problems, suggestions, answers, topics, and users



stackoverflow [azure-devops] Log In Sign Up

Home
PUBLIC
Stack Overflow
Tags
Users
Jobs
TEAMS
Create Team

Questions tagged [azure-devops] Ask Question

Sponsored links for this tag
- Create a free Azure DevOps account today
- Learn more about Azure DevOps
- What is DevOps?
- Free unlimited builds for open source projects on Linux, Mac or Windows
- Migrate from TFS to Azure DevOps

Azure DevOps is a suite of 5 services you use together or independently. For example, Azure Pipelines provides build services (CI), that are free for open source projects and available in the GitHub marketplace. Azure Pipelines also provides release management for continuous delivery (CD) to any ...

Learn more... Top users Synonyms (7)

7,330 questions Info Newest 3 Featured Frequent Votes Active Unanswered

3 votes, 1 answer, 6k views — Create remote GIT branch with Azure Repos
I am implementing a GIT repository in Visual Studio Online (and VS 2015 pro) and I am trying to implement a branching strategy that requires multiple remote branches. So, given that I start with "...
git visual-studio-2015 azure-devops — modified 1 hour ago Quiver 610 ●2 ●12 ●32

1 vote, 1 answer, 454 views — Modify Azure AppService ipsecurity during release from Azure Pipelines
I am trying to add new ip addresses to the whitelist of Azure AppService. I am unable to use XML Transformation or simply replace tokens as the needed list of new entries will be obtained in the ...
powershell azure azure-devops release whitelist — modified 1 hour ago Martin Brandl 34.2k ●10 ●48 ●84

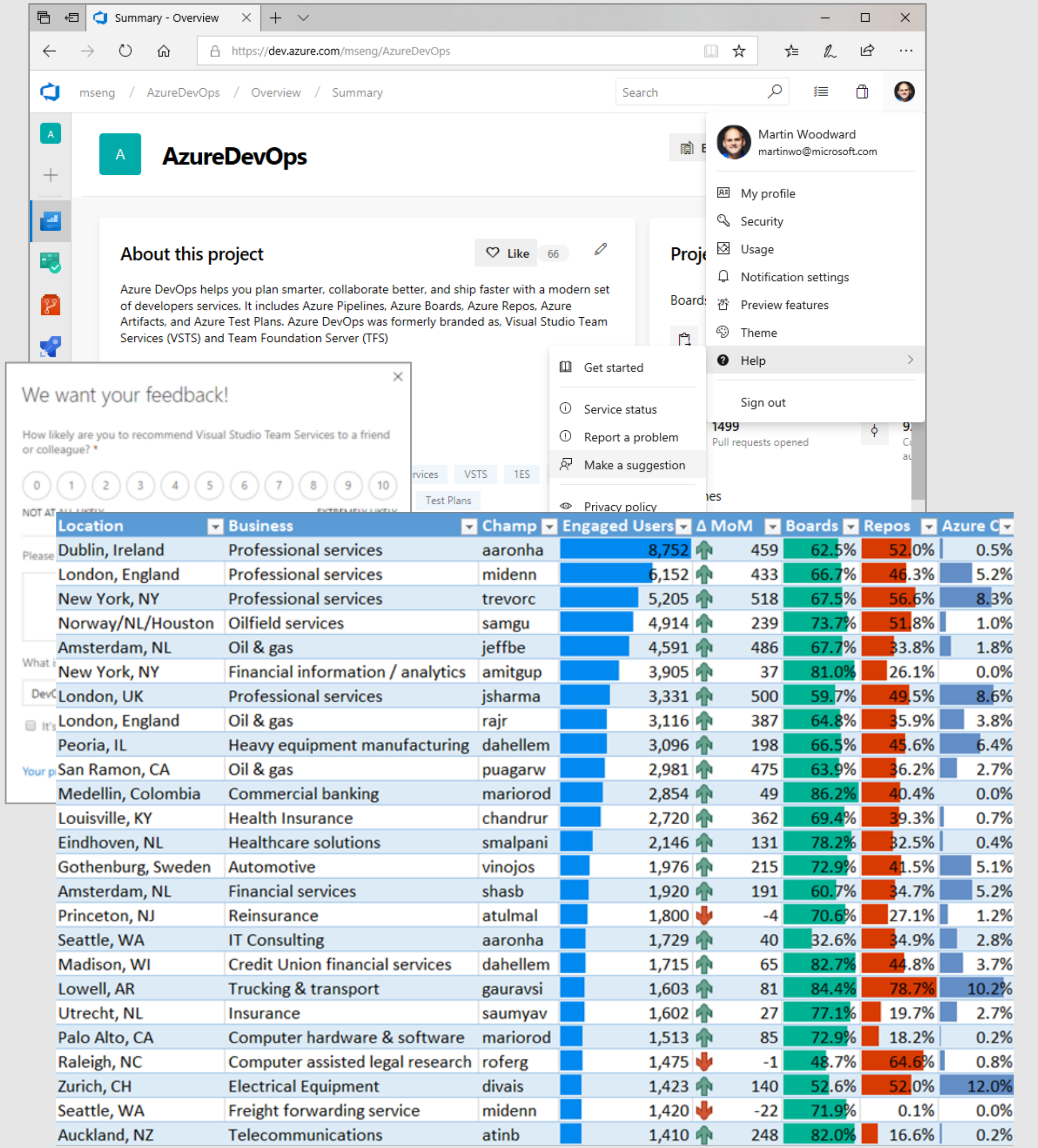
1 vote — Build sqlproj on Azure DevOps
I'm trying to use Azure DevOps Pipelines to build my .NET Core 2.1 solution from GitHub. It includes a SQL

Microsoft Azure
Help secure and manage your Linux or Windows VMs in the cloud
Try Azure for free >

Related Tags
tfs	× 1779
azure-pipelines	× 1020
azure	× 839
git	× 767
visual-studio	× 616
azure-pipelines-release-pipeline	× 470
c#	× 459
azure-pipelines-build-task	× 344
continuous-integration	× 339
tfsbuild	× 334

more related tags

Hot Network Questions
Longest word chain from a list of words
Word for a person responsible for collecting and



Summary - Overview
https://dev.azure.com/mseng/AzureDevOps

mseng / AzureDevOps / Overview / Summary Search

AzureDevOps

About this project Like 66

Azure DevOps helps you plan smarter, collaborate better, and ship faster with a modern set of developers services. It includes Azure Pipelines, Azure Boards, Azure Repos, Azure Artifacts, and Azure Test Plans. Azure DevOps was formerly branded as, Visual Studio Team Services (VSTS) and Team Foundation Server (TFS)

Martin Woodward
martinwo@microsoft.com
My profile
Security
Usage
Notification settings
Preview features
Theme
Help
Sign out

Get started
Service status
Report a problem
Make a suggestion
Privacy policy

1499 Pull requests opened

We want your feedback!

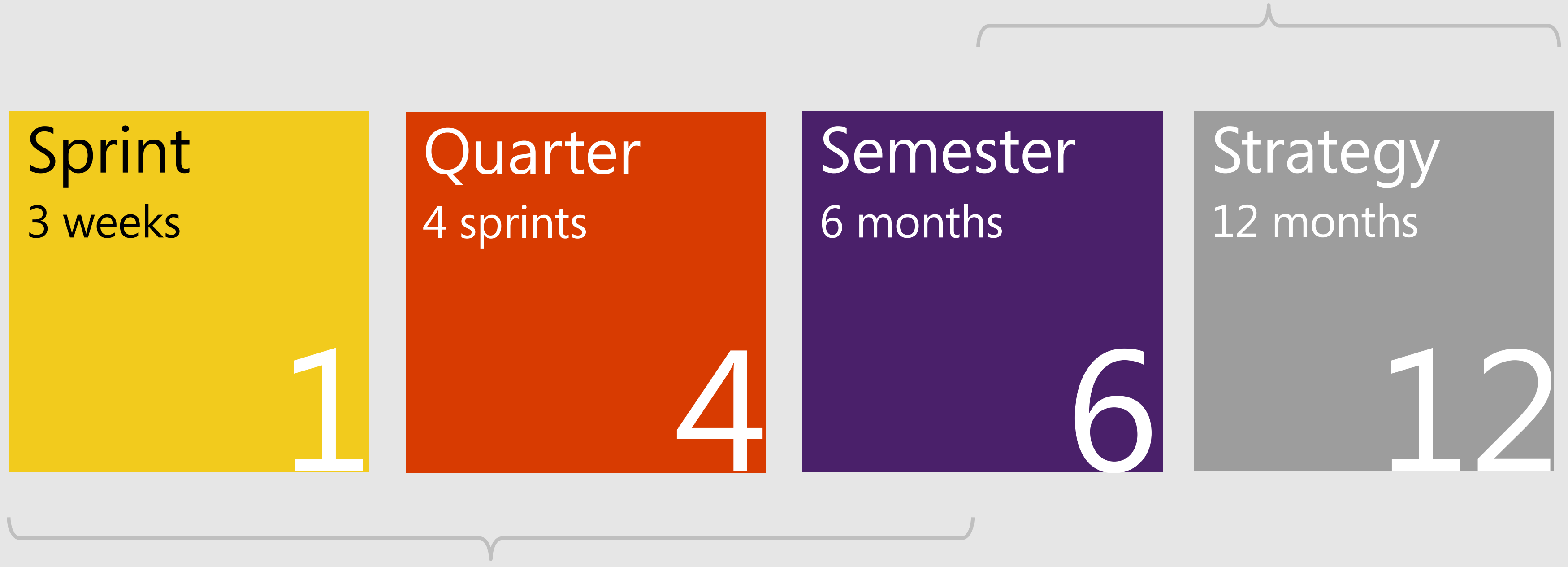
How likely are you to recommend Visual Studio Team Services to a friend or colleague? *

0 1 2 3 4 5 6 7 8 9 10

Location	Business	Champ	Engaged Users	Δ MoM	Boards	Repos	Azure C
Dublin, Ireland	Professional services	aaronha	8,752	↑ 459	62.5%	52.0%	0.5%
London, England	Professional services	midenn	6,152	↑ 433	66.7%	46.3%	5.2%
New York, NY	Professional services	trevorc	5,205	↑ 518	67.5%	56.6%	8.3%
Norway/NL/Houston	Oilfield services	samgu	4,914	↑ 239	73.7%	51.8%	1.0%
Amsterdam, NL	Oil & gas	jeffbe	4,591	↑ 486	67.7%	33.8%	1.8%
New York, NY	Financial information / analytics	amitgup	3,905	↑ 37	81.0%	26.1%	0.0%
London, UK	Professional services	jsharma	3,331	↑ 500	59.7%	49.5%	8.6%
London, England	Oil & gas	rajr	3,116	↑ 387	64.8%	35.9%	3.8%
Peoria, IL	Heavy equipment manufacturing	dahellem	3,096	↑ 198	66.5%	45.6%	6.4%
San Ramon, CA	Oil & gas	puagarw	2,981	↑ 475	63.9%	36.2%	2.7%
Medellin, Colombia	Commercial banking	mariorod	2,854	↑ 49	86.2%	40.4%	0.0%
Louisville, KY	Health Insurance	chandrur	2,720	↑ 362	69.4%	39.3%	0.7%
Eindhoven, NL	Healthcare solutions	smalpani	2,146	↑ 131	78.2%	32.5%	0.4%
Gothenburg, Sweden	Automotive	vinojos	1,976	↑ 215	72.9%	41.5%	5.1%
Amsterdam, NL	Financial services	shasb	1,920	↑ 191	60.7%	34.7%	5.2%
Princeton, NJ	Reinsurance	atulmal	1,800	↓ -4	70.6%	27.1%	1.2%
Seattle, WA	IT Consulting	aaronha	1,729	↑ 40	32.6%	34.9%	2.8%
Madison, WI	Credit Union financial services	dahellem	1,715	↑ 65	82.7%	44.8%	3.7%
Lowell, AR	Trucking & transport	gauravsi	1,603	↑ 81	84.4%	78.7%	10.2%
Utrecht, NL	Insurance	saumyav	1,602	↑ 27	77.1%	19.7%	2.7%
Palo Alto, CA	Computer hardware & software	mariorod	1,513	↑ 85	72.9%	18.2%	0.2%
Raleigh, NC	Computer assisted legal research	roferg	1,475	↓ -1	48.7%	64.6%	0.8%
Zurich, CH	Electrical Equipment	divais	1,423	↑ 140	52.6%	52.0%	12.0%
Seattle, WA	Freight forwarding service	midenn	1,420	↓ -22	71.9%	0.1%	0.0%
Auckland, NZ	Telecommunications	atinb	1,410	↑ 248	82.0%	16.6%	0.2%

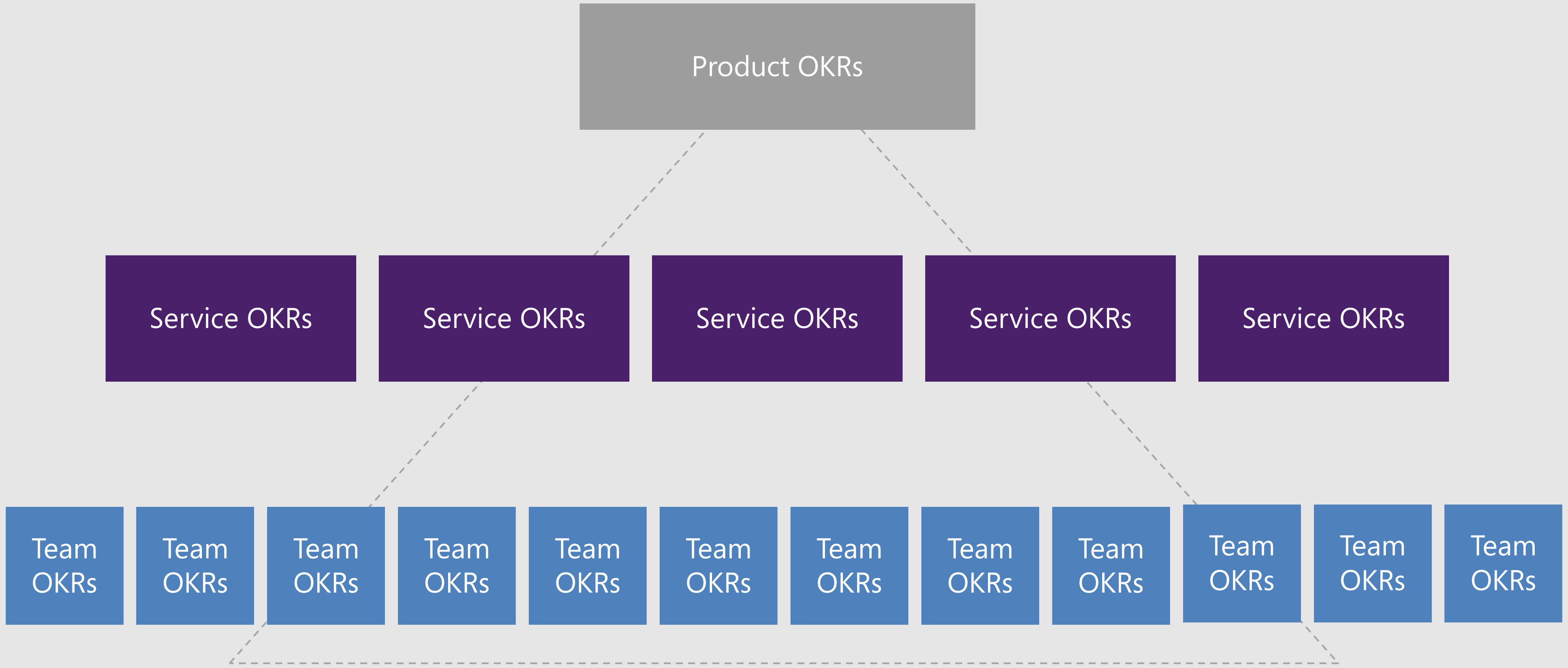
Planning

Leadership is responsible
for the big picture

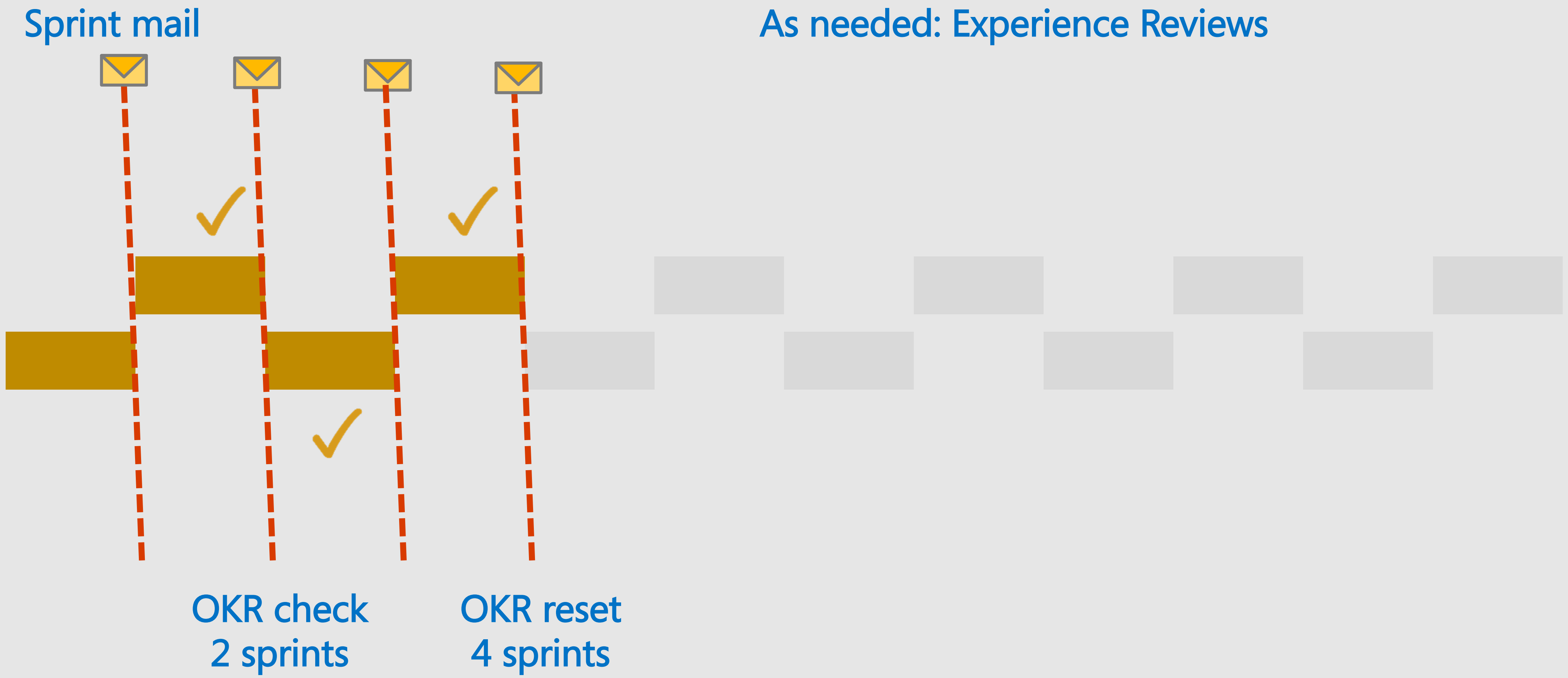


Teams are responsible for the detail

Alignment


Product OKRs
Service OKRs
Service OKRs
Service OKRs
Service OKRs
Service OKRs
Team OKRs
Team OKRs
Team OKRs
Team OKRs
Team OKRs
Team OKRs
Team OKRs
Team OKRs
Team OKRs
Team OKRs
Team OKRs
Team OKRs

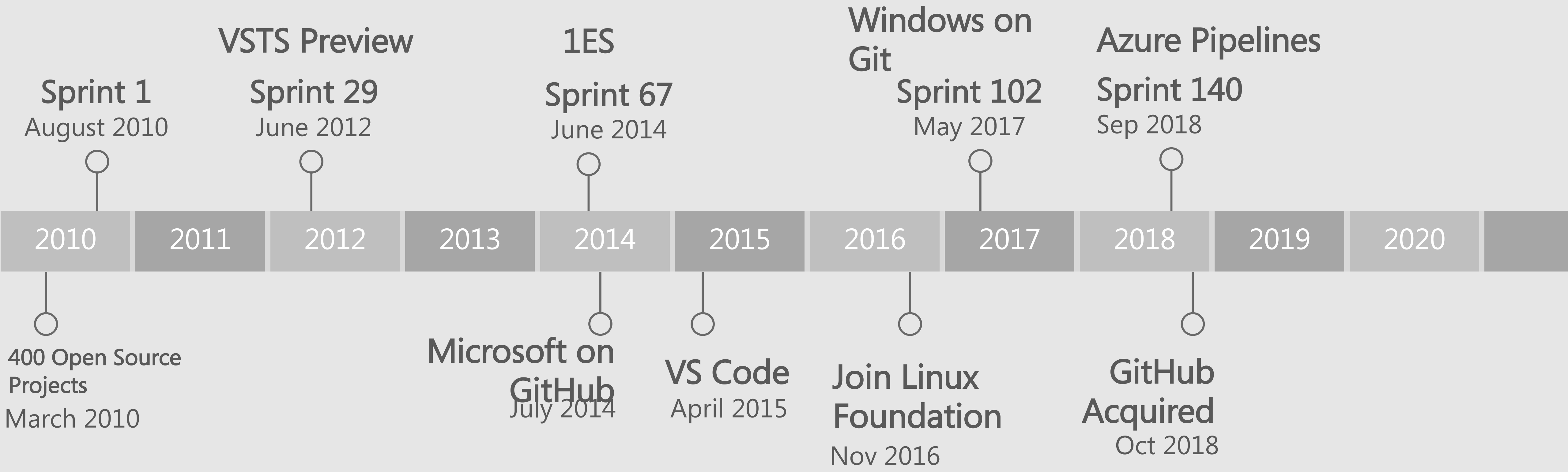
How do you stay in sync?



Microsoft

What’s next?

The journey so far

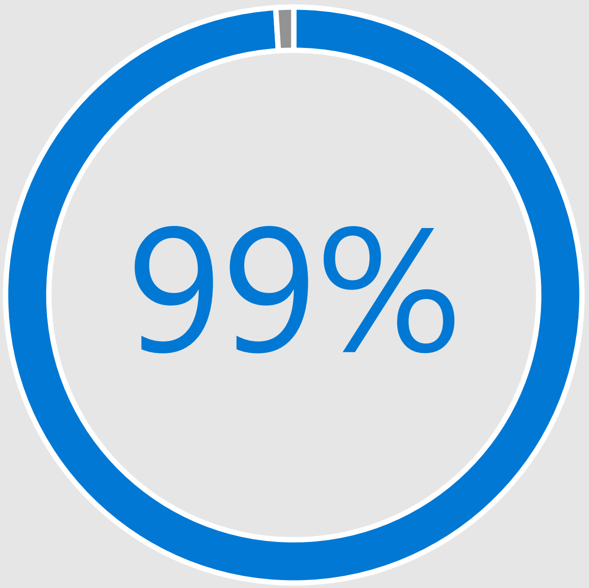


Microsoft

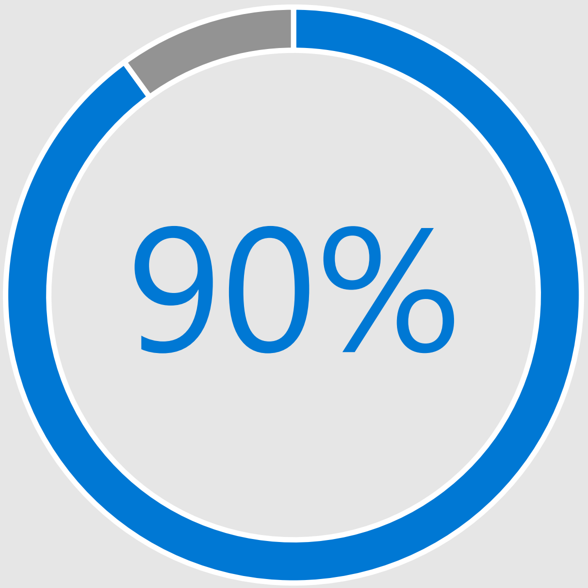
Do you use open source?

Everyone uses open source!

Open Source



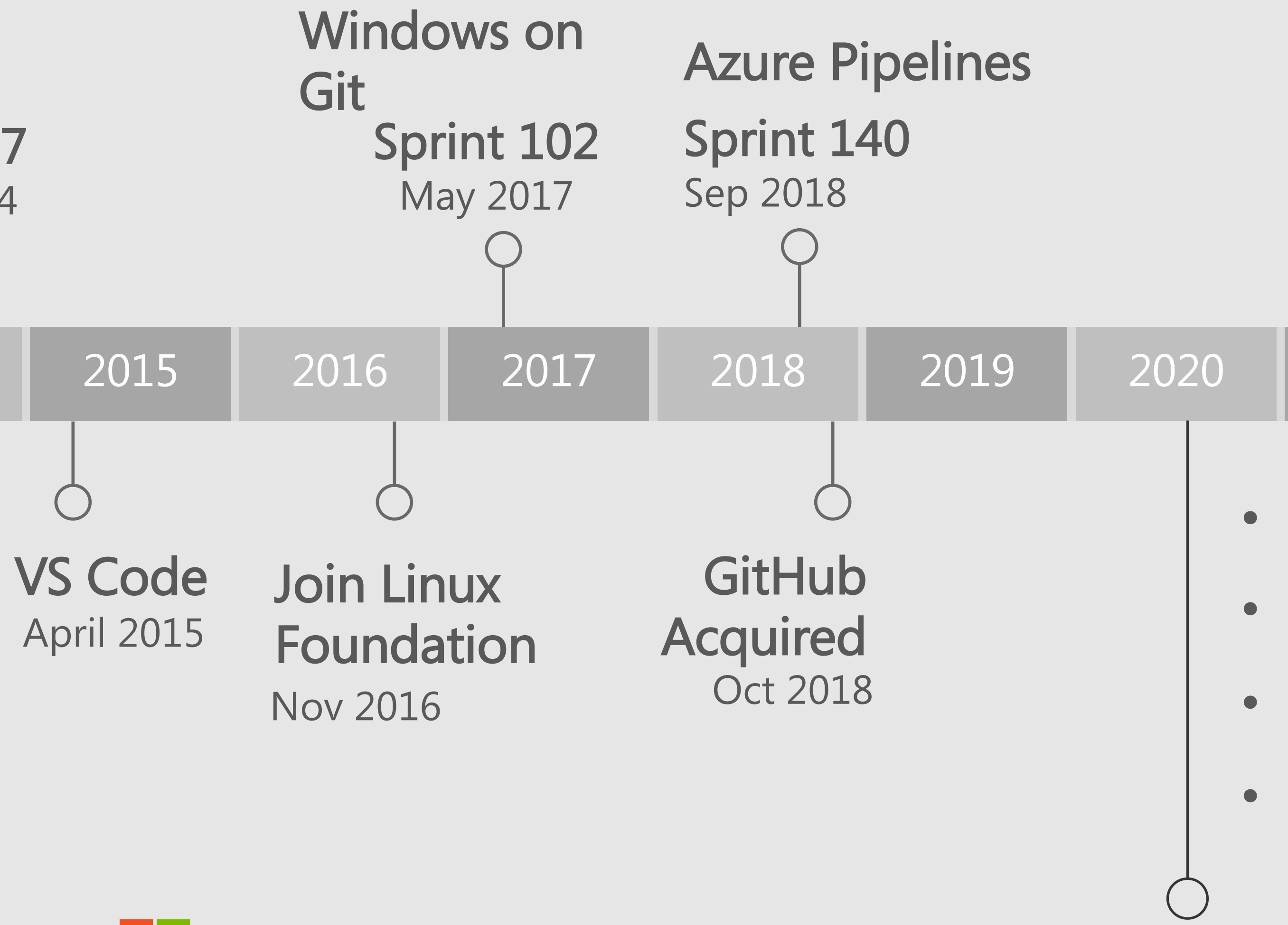
of applications leverage
open source software.



of new code bases are
open source components.

Source: Synopsys OSSRA 2019 report

What’s next?



- Inner source with GitHub
- Shift-left with security & compliance
- Building SRE discipline
- Resilience / Chaos engineering practices

Microsoft

Built with a shared engineering team

Azure DevOps and GitHub share the same leadership

Bringing the requirements and insights of Azure DevOps customers to GitHub

Standardized tooling for 100k engineers at Microsoft

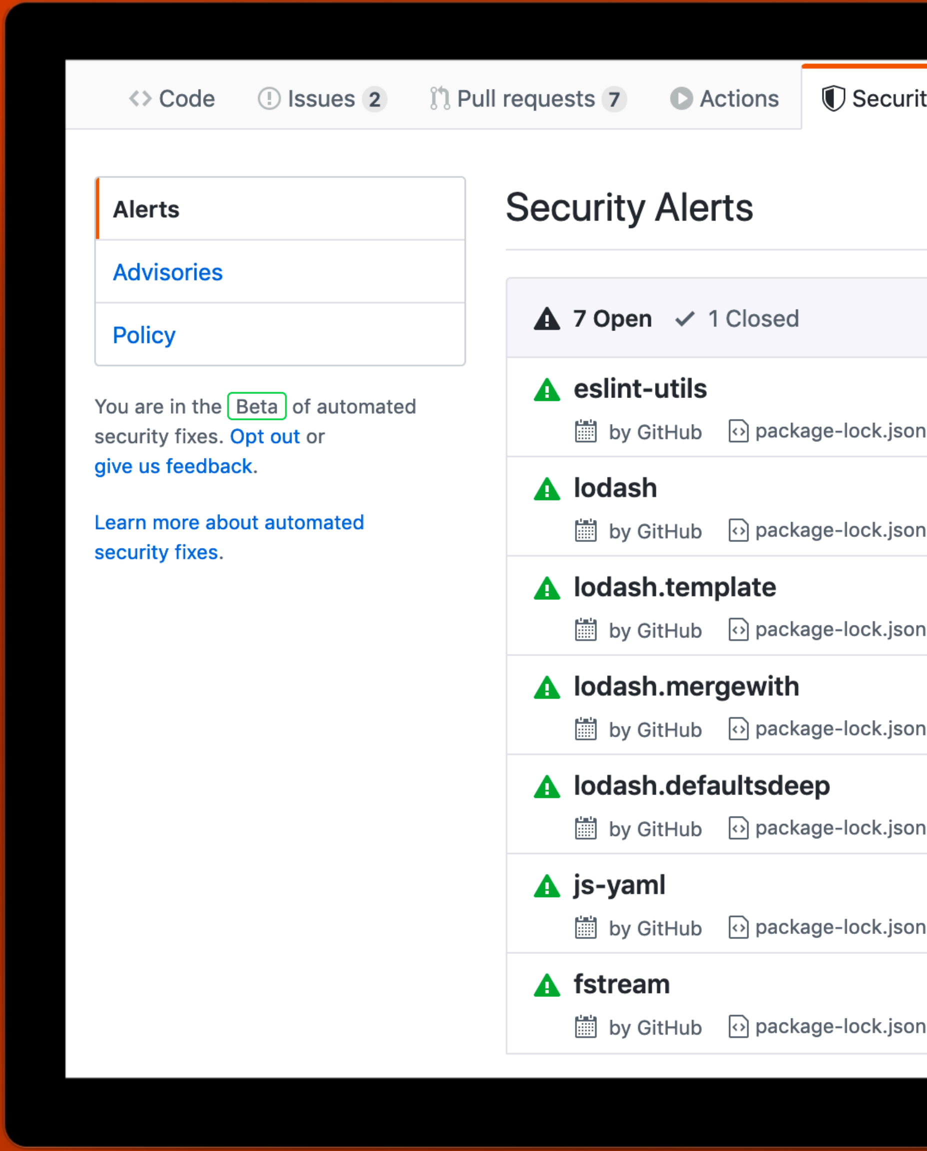
GitHub Advanced Security

Securing the software supply chain

Securing the usage of open source
Vulnerability Dependency Insights and automated security fixes with Dependabot

Pattern based security analysis
Always on security analysis with GitHub advanced security scanning both open source repositories and enterprise code

Global community for security
Integrated into the National Vulnerability Database, MITRE, and WhiteSource for up-to-date security information



Security Alerts

You are in the Beta of automated security fixes. Opt out or give us feedback.

Learn more about automated security fixes.

7 Open ✓ 1 Closed

eslint-utils — by GitHub — package-lock.json
lodash — by GitHub — package-lock.json
lodash.template — by GitHub — package-lock.json
lodash.mergewith — by GitHub — package-lock.json
lodash.defaultsdeep — by GitHub — package-lock.json
js-yaml — by GitHub — package-lock.json
fstream — by GitHub — package-lock.json

Inner source with GitHub

Enabling open source culture and best practices inside your organization

Increased collaboration

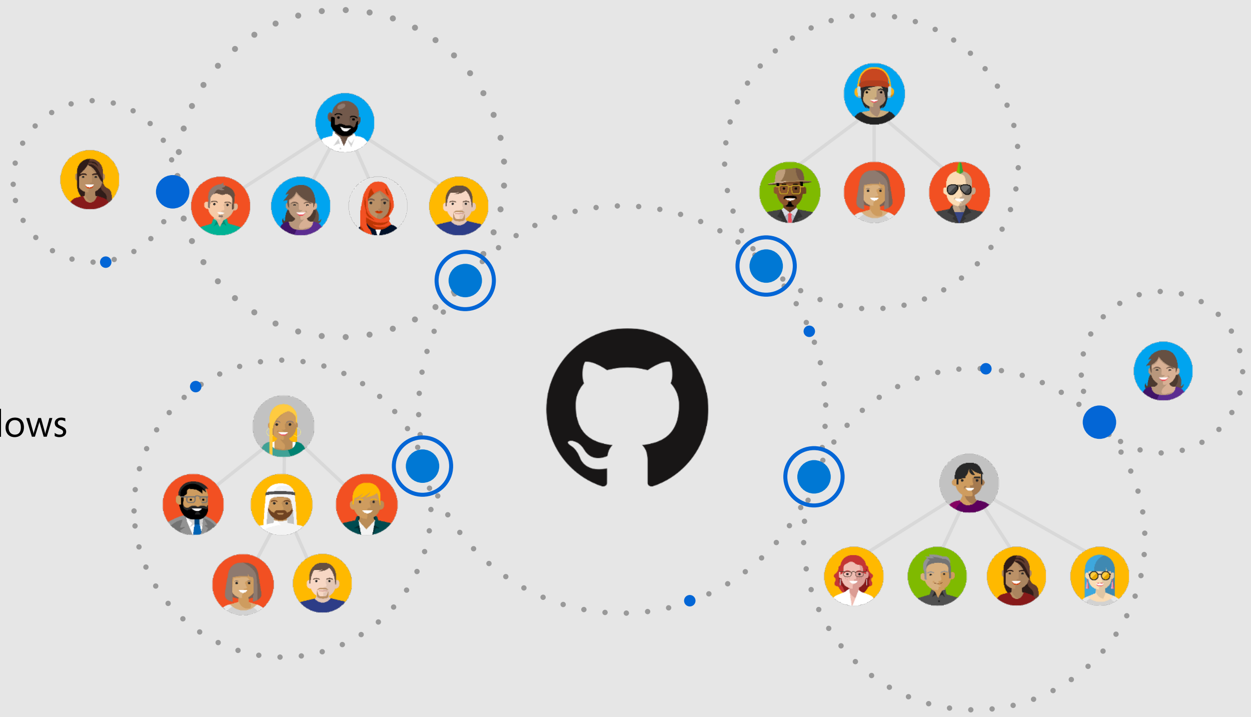
Encourage teams to collaborate within your organization using the same processes and practices as open source communities

Breaking silos

Simplified collaboration across teams, sharing of knowledge, improved code reuse, and secured workflows

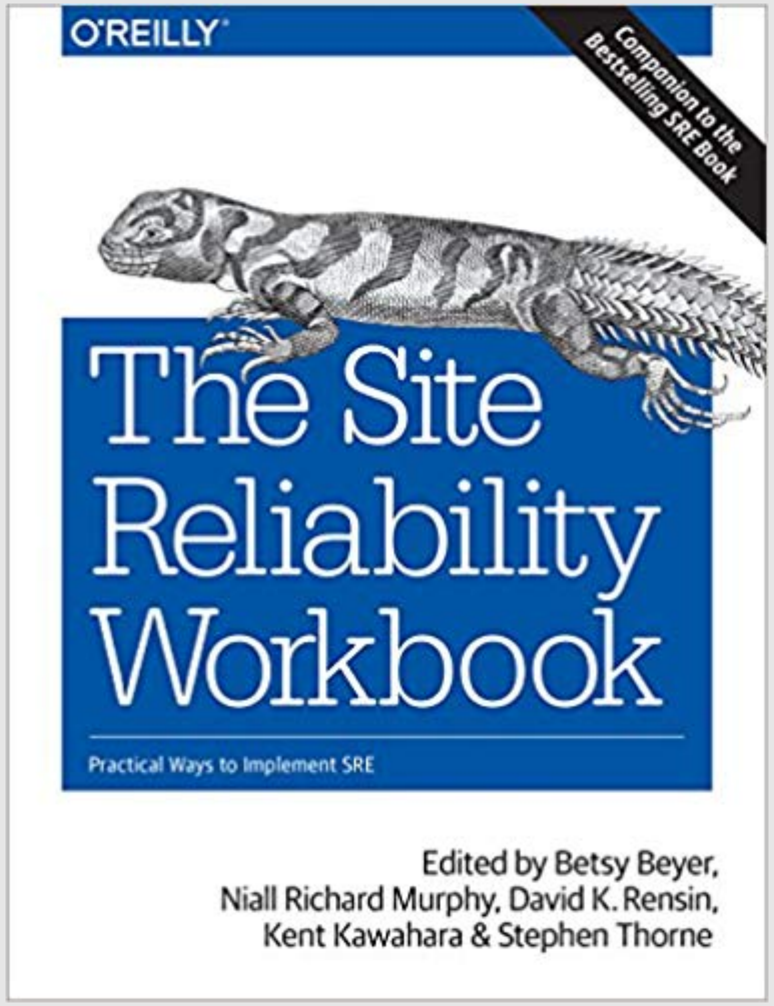
Higher developer satisfaction

Leveraging inner source and open source practices increases developers' satisfaction, enabling them to work on interest projects and increase their skills



Resilience Engineering

- Design with failure in mind
- Circuit breakers
- Self healing systems
- Safely introducing faults to test resilience


O'REILLY
Companion to the Bestselling SRE Book
The Site Reliability Workbook
Practical Ways to Implement SRE
Edited by Betsy Beyer, Niall Richard Murphy, David K. Rensin, Kent Kawahara & Stephen Thorne

- Goal: chaos engineering by default across Azure services

A journey of a thousand miles begins with a single sprint

DevOps is NOT magic!

Thank You!

@DivineOps